# PATENT SPECIFICATION

(11) 1 390 397

1 390 397

(21) Application No. 36455/72 (22) Filed 4 Aug. 1972
(31) Convention Application No. 174 100 (32) Filed 23 Aug. 1971 in (19)
(33) United States of America (US)
(44) Complete Specification published 9 April 1975
(51) INT CL² G06F 15/36
(52) Index at acceptance
G4A 10A 11C 12C 16D 16J 4X 5A 5B 6G 6M1 9C 9D 9F 9X

(72) Inventor RALPH BENJAMIN DeLANO, Jr.

## (54) PROGRAMMED DATA PROCESSING SYSTEM

(71) We, INTERNATIONAL BUSINESS MACHINES CORPORATION, a Corporation organized and existing under the laws of the State of New York in the United States of America, of Armonk, New York 10504, United States of America, do hereby declare the invention for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

Economists have developed the concepts of decision tree analysis, subjective probability, sensitivity analysis and utility theory for use by decision-makers. These concepts have been taught to a substantial number of business decision-makers by such universities as Harvard (Raiffa (10), Schlaifer (11) ), MIT (Kaufman (5), Larson (5) ). Note: These references and others made herein are set out in detail in a bibliography which appears immediately before the description of program subroutines.

The concepts are applicable to a wide variety of problems. Any decision which must be made under uncertainty and where choices are to be made in a consistent manner would appear to be a good candidate for application. Non-programmed decisions in the sense of Simon (14), where the problem is complex and does not occur often enough to justify the computer programming of a specific model, are especially good applications.

Most decisions made by business men are exceeedingly complex. Many variables must be considered. Miller (7) has given evidence that there is a psychological limitation to the number of undimensional variables that the human mind can consider at one time. He proposes several techniques the human mind uses to handle more information at one time. One technique, when one is confronted with a problem having a large number of variables, is to break the task down into a number of absolute judgements.

Decision tree analysis provides a structure whereby a complex problem can be broken down into a number of simpler problems, i.e., determing the different events which may occur, estimating the probability of each, the payoff associated with each event and the inter-relationships between the events. Once this has been done, the format of decision tree analysis provides a consistent method of comparing alternatives each composed of many chance events.

Utility theory provides a technique for reducing unlike variables to a common system of value units. It can also be used to quantify the non-linear relationship existing in a decision-maker's mind relating the size of variable, such as money, to its desirability.

For instance, many small business men do not knowingly run a risk of bankruptcy. even when there is a high probability that the profits will be many times larger than the possible losses. In this sense a lost dollar is not the negative equivalent of one gained. When combined with subjective probability, utility theory allows business men to compare, for example, the spending of money on improvement of employee morale with the spending of money on facilities to increase production.

Subjective probability allows the decision-maker to factor into his systematic analysis the possibility of events occurring about which he is uncertain. A decision-maker is seldom willing to adopt completely the solution from a study which is based on a deterministic model. Little (6) makes essentially this point in his discussion

of a typical use of an oil refinery model based on mathematical programming. The environment of the real world contains many intangibles that are not accounted for in the model. The use of subjective probability in decision-tree analysis allows the decision-maker to include his feelings about these intangibles in a quantitative way and enables him to consider many more intangibles than he does when he based his decision on the combination of judgemental factors and the results of the deterministic model.

Sensitivity analysis may be used to show the decision-maker those subjective probabilities in his situation which are most significant in yielding the results indicated. He can then profitably spend more time considering the probability of the more important events. Sensitivity analysis can also be used to show the decision-maker what would happen if his worst fears regarding a chance event should become a reality.

With the theory developed to this point and a large number of people trained in these concepts, one would expect to find a substantial amountof literature describing their application to many problems. Very little application literature exists. Brown (1) has conducted a survey of the use of decision theory analysis. Raiffa (10) has applied all the concepts except sensitivity analysis to the decisions which occur in prospecting for oil. Schlaifer (12) mentions several business situations. Hespos and Strassman (4) discuss the application of probability distributions to decision trees for investment decision-making in such business fields as new product introduction. Fulton (2) mentions the use of decision trees in new plant location.

There may be reasons why more authors have not written about the use of these concepts. Lack of use is probably the principal reason. Brown (1) supports this conclusion.

Many decision-makers, when faced with a non-programmed decision, are intuitive. They assembled the facts related to the problem and then wait until a bright idea occurs to them. They evaluate this idea and use it if it offers a reasonable solution. Here, "reasonable" means the solution does not contain obvious defects, it satisfies some of their objectives and it contains sufficient novelty to appeal to those most concerned with his solution. The solution is also quite likely to be similar to the decision-maker's previous decisions and to lead to situations which he can control. He is willing to sacrifice return to meet these criteria. An intuitive problem-solver is not likely to use the concepts being discussed.

Others follow more elaborate steps similar to those described by Simon (14) consisting of:

1. Finding, inventing, developing and analyzing alternate courses of action.
2. Selecting the particular course of action from those available in a consistent manner to maximize their return.

Many decision-makers fall into this category, yet apparently the concepts under discussion are not widely used. Little (6) probably has the answer. He says "The big problem with management science models is that managers practically never use them". He goes on to say that much of the problem lies "in the meeting between the manager and the model". Brown (1) discusses the need for a better "package" so that an executive can contribute information and receive conclusions "in a more effective, appealing manner". Sutherland (15), among others, has also discussed this problem. This interface is a substantial barrier for the concepts discussed.

The use of computers having graphic interactive display facilities can reduce this barrier and make the use of decision trees a practical everyday tool for those who have been taught the concepts. Some work has already been done to reduce this barrier. Schlaifer (13) of the Harvard Business School, has written a series of computer programs for the assessment of subjective probability distributions and preference functions. These programs are written in FORTRAN IV and are available from the Business School. Computer programs have also been written to solve decision trees using means probabilities and folding back from the terminal values.

Irwin Miller (8) has described the use of a graphic display for decision making. His work is directed to the solution of marketing and production problems using mathematical equations which describe the relationship between the variables.

According to the invention, there is provided a data processing system including a central processor, a graphic display unit, and programming means which cause the system to generate and display a decision tree, to assign probabilities and expected values to selected branches of the tree in response to input data, to

modify the tree in response to operator selection of branches therein, to calculate modified probabilities and expected values for selected branches of the modified tree, and to display the modified tree and modified probabilities and expected values on the display unit.

An embodiment of the present invention to be described hereinafter may be used to solve the interface problem for a decision maker who understands the concepts of utility theory, subjective probability and decision-tree analysis but has not been able to use them in his decision-making process for one or more of the following reasons:

1. Time required to perform the computations.
2. Paper work required to lay out the tree, assess preference function, etc.
3. Computer professional required between decision maker and computer.
4. Lack of faith in his ability to make assessments of the probability of uncertain events.

The approach is to use an IBM (R.T.M.) 1130—2250 computergraphic display with inputs specified in a problem-oriented language as a tool for decision-making. The computer asks questions of the decision maker in a logical sequence to aid in structuring the decision problems. The questions are phrased in a language which the decision maker understands and he answers by selecting one of the options displayed on the cathode ray tube with his light pen or by typing a few numbers. The computer performs the computations, and the results are displayed on the cathode ray tube. Hard copy output is also provided.

After the tree has been completely specified by the decision maker and displayed, the computer determines the maximum expectation of the proposition and indicates the preferred decisions by showing the path through the tree. At the option of the decision maker, the computer can determine how both the expected utility and the decision path change with variations in assigned probabilities.

The decision maker will see which of the probabilities are most critical and can then concentrate his efforts on making them more accurate. He can, if he wishes, store his results on a disk, leave the computer to obtain additional information and come back to resume the problem where he left off. The computer is available for other tasks in the meantime so the decision maker can feel free of any concern that he is tying up valuable resources while he searches for more information.

The objective is to provide a practical tool for decision makers to use when they must make decisions under uncertainty. The tool does not relieve the decision maker of the need for solid thinking to discover possible events and estimate their probabilities. It can, however, provide structure to his thinking, aid in eliminating bias and inconsistency, enable him to consider many more alternatives and relieve him of computational details. The use of such a tool should lead to better decisions.

In order that the invention can be fully understood, a preferred embodiment thereof will now be described with reference to the accompanying drawings, in which:—

FIG. 1 shows two lotteries having equal expected value and variance;
FIG. 2 shows a preference curve for a conservative decision maker;
FIG. 3 shows a preference curve for a gambler;
FIG. 4 shows lotteries having unequal expected value and variance;
FIG. 5 shows a comparison of preference curves for a conservative decision maker;
FIG. 6 shows two lotteries having equal expected utility and unequal expected value;
FIG. 7 shows a decision tree for new product introduction;
FIG. 8 shows a simplified system diagram of the input section;
FIG. 9 shows a simplified system diagram of output section;
FIG. 10 shows a display of expected value for each decision;
FIG. 11 shows a display for initial problem selection;
FIG. 12 shows a display of complete initial tree;
FIG. 13 shows a magnified section of the tree ready for input data;
FIG. 14 shows a display showing a skip marker;
FIG. 15 shows a display having branches leading nowhere to indicate errors in the assignment of probabilities;
FIG. 16 shows a display of a preference curve and check points;
FIG. 17 shows a sample decision tree;

FIG. 18 shows a display showing results of automated sensitivity analysis on the expected values;

FIG. 19 shows the results of a sensitivity analysis on the first three branches;

FIG. 20 shows the results of a preference analysis using the preference curve of FIG. 16; and

FIG. 21 is a block diagram showing a computer system in accordance with the present invention.

## Description of the Preferred Embodiment

1. Concepts

This portion of the specification discusses the concepts of utility theory, subjective probability, decision tree analysis, sensitivity analysis and those related to the computer such as interactive mode and problem-oriented languages. These concepts have been used in developing the decision maker's tool under discussion. The alternative concept of Monto Carlo simulation which has been used previously in decision analysis is discussed. A comparison from the point of view of the decision maker is made between the chosen concepts and the alternative.

2. Utility Theory

Most people including decision makers in companies have a non-linear preference for acquiring new assets which depends on the value of the asset and the risk involved in acquiring it. For instance, if offered a choice between a certain $500,000 and a 50—50 gamble of winning $1,000,000 or losing $100,000, most people of average means would choose the certain S500,000. They could deposit this money in a savings bank or invest it in bonds and receive a comfortable income for life. The prospects of receiving $1,000,000 would not be a great deal more attractive. In fact, for most individuals, an opportunity to receive $1,000,000 in a lump sum is not worth twice as much as an equal opportunity to receive $500,000. However, the possibility of losing $100,000 which would mean bankruptcy, would be the main factor causing them to choose the certain $500,000, even though the expected value of the gamble is $500,000. But a person who has ample resources to withstand the loss and who likes to gamble might prefer the gamble.

This discussion shows that the desirability or preference for money is not a linear function of the amount of money involved. If one plots preference as a function of amount of dollars, the curve will become concave either upwards or downwards when considering gains, depending on whether the decision maker is a gambler or prefers to avoid risk.

Swalm (16) has shown that combinations of the two shapes also occur frequently. When considering both gains and losses, a conservative decision maker may have a curve that is concave upwards for losses and concave downwards for gains.

The spread of possible returns is one method to account for a decision maker's preference for the certain $500,000 over the gamble illustrated above. A frequently-used measure of spread is the variance. But this technique fails in the case of choosing between the two lotteries shown in FIG. 1. Both lotteries have the same expected value and variance. Many people would prefer the second lottery where they cannot lose any money but would have a probability of 0.1 of winning the terminal value of $10,000.

Utility theory has been developed to account for people's preferences when they wish to be consistent in their choices. A reference lottery is considered where the terminal values are the maximum and minimum amounts to be considered. Preference values are established for assumed amounts between the terminal values by assigning probabilities (P, 1—P) that make the reference lottery equivalent in the decision maker's mind to the assumed amount. The assigned probability, P, then becomes the preference value or "utile" for the assumed amount.

A decision maker may find it easier to determine his preference using 50—50 lotteries. The first lottery is one of a .5 probability of obtaining the maximum amount and a .5 probability of obtaining the minimum amount. The decision maker is asked to express his value of this lottery in dollars. The value is called the certainty monetary equivalent (CME) of the lottery. The preference measurement for the CME is 1/2, since a preference of 1 utile has been assigned to the largest amount and 0 utile to the smallest. The next lottery is a 50—50 gamble on the minimum amount and the CME of the previous lottery. It has a preference of 1/4 utile. The process is continued to obtain additional points on the curve.

A preference curve of utiles vs. dollars as determined by one of the

procedures just described can be plotted for a conservative individual who is considering the equal variance lotteries shown in FIG. 1. It might appear as shown in FIG. 2.

Using utiles in place of the terminal values, the first of the equal variance lotteries in FIG. 1 yields:

$$.9\times.77+.1\times0 = .69$$

and the second

$$.1\times.1+.9\times.7 = .73$$

The second lottery has the higher utility and would be preferred.

On the other hand, the gambler might have a preference curve such as in FIG. 3.

Comparing the utiles for the two lotteries we have:

$$.9\times.3+.1\times0 = .27$$

$$.1\times1.0+.9\times.13 = .22$$

A decision maker who was indifferent to the amounts of money involved would have a linear preference curve. Using utiles for this case would give the same answer as expected values.

Thus, utility theory offers a technique for indicating a decision maker's preference for gaining or losing various amounts of money. It overcomes some deficiencies in the use of either expected value or expected value and variance. It allows a decision maker to separate attitudes toward risk from his estimate of the probability of an event.

There are questions regarding the validity of using a preference curve based on a few lotteries to determine a decision maker's preference for all lotteries within the range of values assessed. For instance, consider assessing the preference curve of FIG. 2 using the second method described previously. The first CME to be determined is for the 50—50 lottery having terminal values of $+10,000 and $—8,000 to determine the value corresponding to .5 utile. This value is $—3,500. A person having a linear preference curve, i.e., one who bases his decisions on expected monetary value (EMV), would judge the lottery to be worth $+1,000. The difference between the two, called the risk premium, is $+4,500. In other words, the CME'er is risk adverse and would rather lose $3,500 than participate in a lottery which, although it has an EMV of $+1,000 also has a .50 probability of costing him $8,000.

Suppose the same individual is asked to particpate in a 50—50 lottery having terminal values of $+1,500 and $—6,500. His CME for this lottery as determined by FIG. 2 is the same, $—2,800. The EMV has been reduced to $—2,500 giving a risk premium of only $1,000. Is the decision maker being inconsistent if he says that, since my possible gain has dropped by a factor of nearly 7 but my equally probable loss is still nearly what it was before, I want more of a risk premium?

Schlaifer (11) has discussed this question of whether such an attitude is a logical inconsistency or the result of non-monetary consequences. The approach taken in this work is to recognize that the condition may exist and to give the decision maker the flexibility to determine its effect on the expectation and decisions for his project.

A technique for dealing with the problem of individual preference has been proposed by Hespos and Strassman (4) who have extended the Monte Carlo simulation technique of Hertz (3) to the sequential decision problem. Their approach is to take a decision tree and replace the branches leading to the terminal values with continuous empirical probability distributions. Monte Carlo simulation is used to generate discrete probability distributions having the same envelope as the continuous distribution. Each decision path is carried all the way through the tree. The decision maker then looks at the various distributions and makes his decision based on their shape.

The choice between distributions is quite burdensome in a complicated situation. There is also the danger of inconsistencies creeping into the choices. Furthermore, since simulations are involved, a substantial amount of computing effort is required to produce the probability distributions.

Hespos and Strassman (4) do mention the possibility of allowing the computer to eliminate some decision paths based on their definition of dominance. "For example, a branch could be eliminated if it had both a lower expected return and a higher variance than an alternative branch". This can be dangerous, as will be shown.

Consider the conservative decision maker discussed previously. Change the second lottery so that he now has a choice between the two lotteries shown in FIG. 4.

If he had only a few thousand dollars in the bank, he would prefer the second lottery, which has a smaller expected value and a larger variance, in order to avoid the possibility of going into debt.

In order to determine the utiles, a new preference curve must be ascertained, since the new lottery is for a larger sum than considered previously. Suppose the new curve turns out to be as shown in FIG. 5, where both preference curves have been drawn to illustrate their similarity.

The new preference curve yields a utility of .63 for the first lottery and .66 for the second. This discussion shows there is danger in using this type of dominance as a basis for automating decisions and that utility theory offers an advantage.

True dominance, where there is no overlapping of the probability density functions, is a perfectly acceptable method to be used in computer decisions. However, the number of instances where true dominance occurs in business decisions under uncertainty will not be sufficient to make it worthwhile.

A change in the type of venture must also be recognized. A decision maker might have entirely different preference curves for a venture into the stock market and one into a horse race.

The use of utility theory enforces consistency into decision making. For instance, it prevents a decision maker from changing his opinion about a version to a specific loss as the amount of money to be gained increases. Consider as an example our decision maker who has the preference curve in FIG. 2. He would be indifferent to the two lotteries shown in FIG. 6, in spite of the fact that the maximum amount of money to be gained in the second case is larger. The probability of the increased gain is less so that the expected utility of both lotteries is the same.

Note that the conservatism of the decision maker, as expressed by his preference curve, has caused him to attach no increased value to a lottery having an expected value which is larger by a factor of more than three. If the decision maker does not feel he would actually consider the two lotteries to be equal, then he is either being inconsistent or his preference curve should be revised. A preference curve should be tested after the original assessment by situations such as these to insure validity.

3. Subjective Probability

Subjective probability may be defined as a measuie of the decision maker's degree of belief of the occurrence of an event as evidenced by his action pattern.

Many decision makers who have not been trained in subjective probability assessment techniques do not believe in using probability to make decisions. One of their main arguments is that they cannot assign a number to a chance event about which they know very little.

One of the premises of the present approach is that a decision maker, when making decisions under conditions of uncertainty, should use all the significant information available to him. For instance, he should not neglect to assign a probability to an event which he believes will produce an appreciable effect on the outcome of his project just because he finds it difficult. In the process of using the decision tool disclosed herein, he will find out those probabilities which can affect the decisions or values materially and those which need not be considered further. The decision maker may then reassure himself by searching for more information about the significant probabilities with some assurance that he is not wasting his time.

If the decision maker is not willing to use all the information he has, i.e., assess probabilities of uncertain events, and if this information is significant, he probably will make the wrong decision. In effect, he has assumed that the event will not occur. Yet he knows that there is a possibility that it will. Hence, he is being inconsistent in his decision-making and should assign a probability to the event.

The decision maker must exercise caution in distinguishing between the value of an event and the probability of that event. He must not account for uncertainties in both items. For instance, if he assigns what he believes to be a large value to an event he should reflect this doubt in the probability of that value and not account

for his uncertainties twice by reducing both the value and the probability in an unrelated manner.

4. Decision Tree Analysis

A decision tree is made up of alternating decision and event nodes interconnected by the branches. Each node has only one branch on the left side but many have any number of the right. The probability of an event is noted by assigning a probability to that branch. The sum of the probabilities for all the branches leading from a chance node must be one. The number of stages in a tree is the number of nodes found along the path composed of branches which has the maximum number of nodes.

The last stage in the tree is composed of event nodes and the branches leading from these nodes end in terminal values. These terminal values are the payoffs for the branch associated with that event. A simple decision tree taken from Hespos and Strassman (4) is shown in FIG. 7.

The optimal sequence of decisions in a tree is found by a technique called "averaging out and folding back". Starting with the terminal values associated with a node, the expected value for that node is determined by taking the sum of the products of the terminal values and their associated probabilities. The maximum expected value is chosen for each decision node and the process repeated until an expected value is obtained for the first decision node. This is then called the value of the proposition and the chosen decision branches represent the optimum decision.

The concept of decision tree analysis provides a systematic approach to many management problems. The format enables a decision maker to break down a complex problem into a number of simpler ones. According to the thoughts of Miller (7) this should enable a decision maker to consider more possibilities. In addition, following the decision tree format may suggest additional possibilities that should be considered. The problem of handling too many alternatives will be discussed in the next section.

5. Sensitivity Analysis

Sensitivity analysis is commonly used in operations research to determine the effect changes in input parameters have on the proposed solution. In the present case, the decision maker is understandably concerned regarding the accuracy of the probabilities he is using in his decision tree analysis.

Two types of sensitivity analysis have been implemented in the work under discussion. An automated sensitivity analysis determines which probability along the preferred decision path is most sensitive in changing the path and which is most sensitive in changing expectation. This type of sensitivity analysis is limited to determining the effects of changes in probability close to the value the decision maker has specified.

If the decision maker suspects that he may have made larger errors in assigning some probabilities, especially those not on the decision path, he will want to determine the effects of changes in those probabilities. He may select any probability on the tree, change it and the computer will determine the new expectation and decision path. A decision maker would also use this type of sensitivity analysis to determine if he should reject an undertaking because of difficulties in assessing one or more probabilities. Sensitivity analysis can tell the decision maker which of these probabilities:

1. Is most critical in obtaining the previously computed expectation of the proposition.

2. Is most critical in changing the decisions required to obtain an optimum expectation.

3. Is not significant in determing either expectation or decision path. The decision maker can prune branches containing such probabilities and, if he wishes, add additional alternatives.

Once the decision maker knows what probabilities are most critical, he can then concentrate his efforts on revising them without feeling he may be wasting his time on inconsequential factors. Or, he may choose a set of decisions which, while less optimum, have a smaller probability of producing a loss.

In effect, sensitivity analysis, when combined with decision tree analysis based on using utiles instead of dollar value, provides the decision maker with results which are similar to the stochastic decision tree analysis of Hespos and Strassman (4). Since only a few changes in probability need be made instead of generating changes at random, results can be obtained in a much shorter time than with simulation. Substantial savings in the decision maker's time are possible. The

analysis can be made more flexible and can be tailored to fit specific situations. The decisions will also be more consistent, as mentioned in the section on utility theory.

6. Problem-Orientated Languages and Interactive Mode

These two subjects are combined since they are closely related. Their use here stems from the objectives of this invention. This discussion begins with a restatement of the objectives and the strategy for attaining them.

The system described in detail hereinafter provides the machine portion of a man-machine system wherein the allocation of tasks is such that each of the two components does those tasks for which it is best suited. The man supplies the creative ability, judgement, imagination, intuition and experience in making the unformatted decisions. The machine performs the computation, makes decisions where explicit rules are provided, provides consistency checks, and presents the results to the man in a format which not only can be understood but which may suggest possible new approaches. It should do all this in a manner which will enhance the man's abilities as much as possible.

This division of effort demands the use of a problem-orientated language with the computer operating in an interactive mode. These two rather recent developments in computer technology provide the communication which links the man with the machine to provide a system.

The historical sequence of the development of computer languages was: machine, assembly, procedure-oriented and problem-oriented. Each language in the hierarchy simplified the knowledge of computer operations required of the user, usually at the expense of less efficient use of the computer. The tremendous decrease in cost for a computer to perform an individual operation, however, has made it possible to use problem-oriented languages and thereby improve the overall efficiency of man-machine systems by several orders of magnitude.

Problem-oriented languages enable the user (as constrasted with having to engage that intermediary—the programmer) to converse with the computer effectively. They contain commands for the computer to perform a function in a form so as to be easily understood by the user. Their use requires no conventional programming experience.

The trial-and-error process is different from the usual mathematical optimization technique. The data is dynamic and the steps required to build up a solution are unpredictable. Each step depends on the results of the previous steps and the results may also make it desirable to alter the input data. The computer program must be written in modular form with the decision maker able to call for the modules in any sequence. Each module performs a basic operation.

Operating a computer having graphic capability in an interactive mode is the other portion of the essential communication link. Batch processing is very unsuccessful for solving problems by a trial-and-error process unless only a few alternatives are to be evaluated. It is too inflexible, processing time takes hours and sometimes days and many batch processing facilities require intermediary facilities to prepare the input, sequence a number of jobs and manage the compute operations. In an interactive mode the decision maker has "hand-on access" to the computer with no intermediary and with the computer providing, hopefully, instantaneous response to his commands. He can build up his decision in increments, using the computer continuously as he shifts his approach in an unstructured manner.

The graphic capability is an essential portion of the communication link since it changes strings of data into patterns and allows the decision maker to see his problem more clearly. Output speed is not limited by the requirement for mechanical motion and the decision maker can select program alternatives with the stroke of a light pen. He can also use the light pen to select values he wishes to change. Most executives resent having to type. The reduction in the use of a typewriter made possible by having light pen capability will help overcome this psychological barrier.

The combination of a problem-oriented language and a display operating in an interactive mode can reduce much of the frustration in supplying data to a computer. The following principles are desirable to simplify the input procedure:

1. Minimize the effort required of the user in providing data.
2. Provide correction, automated checking and interrupt features.
3. Provide patterns showing the interrelationships between elements of data.
4. Show how the computer has interpreted the data.
5. Display adequate instructions.

7. Implementation

The purpose of this invention is to provide a decision maker who has had previous training in the use of decision tree analysis, subjective probability and utility theory with a tool to use in making decisions using decision tree analysis. The tool is designed to reduce some of the man-machine interface problems which occur when a computer is used to perform decision tree analysis. The preferred approach is to combine a small computer having graphic display and light pen selection capabilities, with a problem-oriented language and a flexible program, in a manner to facilitate operating in an interactive mode.

Such operation is fostered by display of the interrelationships of the supplied data, selection of program alternates by light pen and rapid computer response. Error detection is provided to simplify the process of supplying data. Both an automated and a selected sensitivity analysis are provided to enable the decision maker to answer "what if" questions regarding the effects of changes in probability on expectation and decision path. Choices at the decision nodes are selected by the computer. The selection is based on maximizing expected value or expected utility. In the latter case a preference function supplied by the decision maker is used. Guides are provided for the decision maker to use in assessing his preference function, and the effects of changing the function may be determined. Sufficient flexibility is provided to allow the decision maker to structure and solve his problem using a trial-and-error technique. No previous experience with programming is required.

This chapter discusses those aspects which are believed to be most significant in achieving the objective. The hardware is discussed first, followed by a section describing the program functions. Sections on the input procedure and sensitivity analysis follow. The final section discusses the solution of a hypothetical decision problem involving the drilling of an oil well. At the end of the specification, there are a list of subroutine functions and a list of the complete program with explanatory comments.

8. Hardware

Referring to FIG. 21, there is disclosed an IBM (R.T.M.) 1130 Model 3D computer having a central processing unit 1. A 2250 Model 4 graphic display 2 was chosen for implementation. The computer has 32,768 sixteen-bit words of core storage in memory 5 and operates with a storage cycle time of 2.2 microseconds. Both the storage capacity and cycle time are quite adequate. A total of 25,000 words of core are used. A computer having a smaller direct access memory (16,000 words, for example) could be used at the expense of waiting several seconds at some point in the program execution for the memory to be erased and reloaded. Previous experience with an 1130 computer having only 8,192 words of core memory indicated that this size memory was inadequate. The initial input and graphic subroutines require more than 9,000 words. Disk storage 6 is used for the mainline program, the subroutines, the data for 16 problems and the FORTRAN compiler.

The CRT graphic display 2 has a usable area of 12 inches by 12 inches, although this is considered by the experimental graphic subroutine package to be a 10-inch square. The smallest single step is 0.01 inch giving a roster of 1,000 by 1,000 available elements. The smallest character which can be displayed using the subroutine package is 0.14 inch by 0.14 inch. Allowing a minimum spacing between characters, only 66 branches of a decision tree can be labeled. Consequently, the tree has been split into four vertical sections during the input phase, as will be discussed in the section on input procedure. The 2250 display unit 2 can generate a complete display in a few seconds. Regeneration of the display is automatic.

A fibre-optic light pen is provided with the graphic display 2. This allows the user to select previously-designated alternates in the program. The input section describes the use of the light pen.

A group of pushbuttons called function-keys is also available on the function keyboard 10 for making predetermined modification to the program. Lights associated with each function-key indicate whether the key is on or off. Two of the keys are used, one to advance the program to the next step and the other to enter zeros for all the terminal branches of the tree associated with a node.

Three methods are available for supplying hard copy output of the display. The IBM (R.T.M.) Model 2285 display copier 8 may be used to provide 8-½ by 11 inch paper copy of the associated 2250 display 2. The experimental graphic subroutines which are used make use of a copying feature to duplicate the display on the Model 1627 plotter 4. The plotter 4 has X and Y increment speeds of 18,000

1/100-inch steps per minute and a pen status change speed of 600 operations per minute. Several minutes are required to draw a complete display. The method favored by the author is to photograph the display screen, using a camera with a ground glass back for focusing and a Polaroid attachment which uses 4 x 5 inch type 52 Land film. Exposures of about 1/25 second at a lens opening of f8 have given satisfactory results with the display beam intensity set at a moderate level. A faster exposure should not be used since complete regeneration of the display may require 1/30 second.

A conventional mechanical printer 7 is used to provide a list of the decision branches selected and the expectation each time these are computed.

The 2250 has an alphanumeric typewriter keyboard 9 on which the user types input values and probabilities. Its use is covered in the input section.

A standard card reader 3 is used to read three cards required to start the program. The mainprogram and sub-routines are stored in core image format in order to speed the transfer of the program from disk to main memory. The waiting time until the first image appears on the screen is only 12 seconds. This is the longest wait for a computer response in the complete program.

9. Program Functions

This section gives an overview of the functions available to the decision maker The functions are shown in flow diagram form in Figures 8 and 9. A more detailed description of the operating procedure will be found in the next section.

The first display consists of a list of the titles for as many as 16 previously-stored problems and the word "new". The decision maker selects either one of the stored problems or "new". If he selects one of the problems the tree is displayed with the previously stored data positioned on the proper branches. He may then proceed to revise his values or add more data by selecting "correct". He may instead pass on through "calculate" to perform a sensitivity analysis or assess his preference curve.

When "new" is selected the complete tree is displayed with a series of instructions and the decision maker is asked to select a section of the tree so he may start supplying data as described in the next section.

After all the data has been furnished the decision maker selects "calculate" to display the expected value. The computer adds the products of each terminal value and its associated probability for each individual node. The largest expected value for each third-stage node is selected and the averaging process is repeated using the second stage probabilities. Selection of the largest of these product sums yields the expected value of the project. This value as well as the selected third-stage values, are displayed on the tree as shown in FIG. 10. The corresponding branches are broadened for easier identification.

The decision maker may, if he wishes, select "correct" to eliminate some of the branches. This will provide space for adding new branches representing additional alternatives that he wishes to evaluate. The decision maker also has the option of selecting "preference", "sensitivity", "store", "correct", "fini", or "vary prob". Selecting "preference" allows him to determine his preference curve as described in the next section. After this has been done, the computer converts the terminal values into utiles using linear interpolation between the assessed points. It then averages out, folds back, computes the expected utility of the projects and displays the results as before. Before the results are displayed, they are converted back into quantities having the same dimensions as those used for the terminal values. At the end of the preference analysis, as for any of the functions in this section, the decision maker may select from any of the options listed above.

The functions "sensitivity", "store", and "correct" are described more fully in the next sections. Sensitivity analysis is performed using expected values it it is selected before "preference" has been selected and expected utiles afterward. The decision values displayed always have the same dimensions as those supplied for the terminal values, with the exception of a possible scale change to thousands as discussed in the following section.

10. Input procedure

This section describes the procedure used by the decision maker in supplying the data required. The first subsection describes how terminal values and probability are supplied. The second describes the assessing and checking of the preference functions needed for decision tree analysis based on expected utility.

Terminal Values

The decision maker is expected to lay out his problem in a four-stage, four-branch-per-stage decision tree format, and assign probabilities to chance events

and terminal values before he starts using the computer. A form can be used to facilitate this procedure.

The program is stored in disk storage 6. After a few cards have been read by the card reader 3, the display offers the user a choice of 16 previously-stored problems as shown in FIG. 11. The decision maker may select one of these or he may select to start a new one.

Selection is by light pen as it is for all program alternatives. The experimental graphic subroutines provide for brightening the item under the light pen before the pen is pushed to make a selection. This provision reduces the possibility of a selection error.

If the decision maker wishes to start a new problem, the initial tree is displayed as shown in FIG. 12. A series of instructions appear at the bottom of the display describing how the information is to be supplied.

Hardware limitations discussed previously make it impossible to display all possible terminal values for a complete tree opposite the branch with which they are associated. The tree is split into four sections during the input phase, and only one section is displayed at a time. The section being displayed is indicated by the arrangement of branches at the first stage node as shown in FIG. 13.

The decision maker uses the keyboard 9 for numeral data only. The proper branch number of the tree is given with each request for data. As this data is supplied it appears on the bottom line of the display, and an underline cursor indicates that the computer is waiting for a character and its location. An error may be corrected by using the backspace key to move the cursor under the number to be replaced. Typing the new number causes it to replace the old. K and M are used to represent thousand and million to avoid having to key strings of zeros. After the last character the user pushes the alternate coding and ends keys simultaneously to start the checking operation.

The program scans each value, looking for illegitimate characters. Only the numerals 0 through 9, blanks, decimal points, dollar signs, plus, minus and K or M are allowed. An error message is displayed when any other character is used, and the user is asked to repeat the data. An error message is also printed it more than three digits are used. This requirement makes it possible to avoid using E format (floating point), which would be required otherwise because of hardware limitations within the computer. E format, in which a number such as 62,500 is designated by .625 E5, is not familiar to many people. Imbedded blanks, dollar signs and plus signs are ignored. A K causes the value to be multiplied by 1000. When the first M is used, all previous values in storage are mulitplied by 1-1000 and an appropriate message is displayed. After these checks are satisfied the value is displayed at the end of the proper branch of the tree. A marker is also displayed after the first value containing an M, to remind the user of the scale change.

The sum of the probabilities for each node is checked to ensure that this sum is either zero of between 0.9 and 1.1. When the sum is outside this range, an error message is displayed telling the user what the sum is and asking him to repeat all the data associated with the node. The figures are also erased from the appropriate branches of the tree. This checking feature is used for probabilities in place of the one used for values.

Many problems will use only a few of the terminal branches. Consequently, three provisions have been made to reduce the effort required of the user in entering values for unused branches. Simultaneous pushing of the alternate coding and end keys on function keyboard 10 enters a zero value or probability for a branch and advances the program to the next step. The second provision is to use a function key to enter zeros for all of the branches of a terminal node at one time. A marker is displayed to indicate those nodes for which all zeros have been entered and which branch is to receive the next input, as shown in FIG. 14. The menu displaying section numbers is always available to allow the user to skip to a new section. This provision makes it possible to skip the unused branches which are below the active portion of a section.

If the decision maker spots an error in the data he has entered previously, he may select that value or probability with a light pen and the program will delete all the data associated with the parent node and ask for new data. The data are checked as before, and the program resumes at the point where the interruption occurred.

In order to overcome some of the limitations resulting from a limited portion of the tree being displayed during the input of terminal values, provision has been made to display the complete tree with unused terminal branches omitted. The user

selects "all sections" with the light pen. The tree is displayed after he has provided the next terminal value, which can of course be zero. This provision allows the user to see the interrelationships in decision-tree format among all the data he has supplied and plan for the location of future data. He may then select "correct" to change previous values or continue on with providing terminal values and their probabilities.

The decision maker may interrupt his work at any time by selecting "store" from the menu with the light pen. He might wish to do this because of the pressing demands for his attention elsewhere or because he needs time to think about his problem and perhaps obtain more data. When he returns to the computer, the tree is displayed, showing the data he had supplied previously, and he can continue from the point of interruption.

The description thus far has covered the handling of values and the probabilities for the terminal branches. Probabilities for the second stage are handled in a similar way.

After all the data are supplied, the complete tree is displayed, as shown in FIG 15, to show the interrelationships of all the data. Terminal branches (i.e., fourth stage) having zero probability are eliminated to reduce confusion and provide more space for displaying the terminal data. Third-stage branches are eliminated when the corresponding second-stage probability is zero.

The elimination of the third and fourth-stage branches also enables the decision maker to spot errors in the assignment of data. If a probability has been assigned to a second-stage branch but not for any of the corresponding terminal branches, all the branches leading from the second stage node are shown, but no connecting terminal branches appear. Third-stage branches are left hanging in mid-air when the corresponding second stage branch probability is zero. Errors have been introduced in the data of FIG. 15 to show both conditions. These features show the decision maker that he has not matched second and third-stage branches.

Preference Assessment

The method of assessing the decision maker's preference curve is similar to that described by Schlaifer (11, 13). The decision maker is asked first to give his certainty monetary equivalent (CME) for a 50—50 lottery involving his maximum and minimum values. This establishes the .5 utile value. Two similar 50—50 lotterie combining either the maximum or minimum terminal value and the previously assessed CME determine the values corresponding to .25 and .75 utiles.

A curve is displayed drawn through the three assessed points and the maximum and minimum terminal values. Each CME is checked using the same input subroutines as was used with the terminal values. Display of the curve gives the decision maker a further check in avoiding possible errors in providing the data.

The decision maker is provided with two techniques to assist him in eliminating inconsistencies in his preference function. He is asked to compare his curve with the straight line representing the preference curve of an individual who is willing to accept expected monetary value as a basis for decision-making. If his three assessed points lie on both sides of the straight line in a random fashion for instance, he may wish to reassess the function.

The second technique is to have the decision maker give his CME for two additional 50—50 lotteries. The two values for the first lottery are the previously-assessed values corresponding to .25 and .75 utile, and for the second lottery the .25 value and the maximum value are used. If the utile equivalents for these values do not lie on or close to his preference and the decision maker believes this represents an inconsistency, he can repeat the assessment procedure. A completed preference curve with checkpoints is shown in FIG. 16.

Since the shape of the preference curve can affect project expectations substantially, the decision maker may wish to try several curves. The program allows him to repeat the assessment procedure and determine the expectation of his project for as many different curves as he wishes. He may wish to try a second curve, for instance, when he believes he is being consistent in spite of the fact that the points for the two check lotteries do not lie on his preference curve.

11. Sensitivity Analysis

Sensitivity analysis is performed after the terminal values have been averaged out, folded back, the project value computed and the decision path determined on the basis of maximum expectations, i.e., either maximum expected value or greatest expected utility.

The automated sensitivity analysis assumes that the decision maker wants to

know how sensitive his decisions and expectations are to small changes in those probabilities that lie on the decision path. It assumes that he is more concerned about a reduction in project value than an increase. Two probability sensitives are computed, one for decision change and one for expectation change.

The probability which changes the least in causing a decision path change to the next lower expectation is selected for the decision change. The previous assumptions allow this probability change to be used in determining the probability most sensitive in changing the expectation of the project. The ratio of change in expectation to the change in probability computed for a decision change is determined for each decision node. Since expectation is a linear function of probability up to the point of a decision change, this ratio is independent of the change in probability. The probability producing the largest ratio is selected as the most sensitive in producing an expectation change.

The previous assumptions also provide a criterion for determining the reduction in expectation used to measure probability sensitivity. This criterion is the minimum expectation change which causes a change in the decision path. If an increase in expectation were chosen the criterion would be more arbitary.

At each decision node, provided that the chance node following the branch having the largest expectation has two or more branches, those branches following the chance node and having the largest and smallest expectation are determined. The probability of the branch having the largest expectation is reduced and that having the smallest expectation is increased. The increase is just sufficient to cause a previous branch to have an expectation equal to that of the next most favorable decision. This change in probability is called the decision probability sensitivity for the node under consideration. The process is repeated for the branches following along the decision path. The probability most sensitive in producing a decision change for the tree is selected as the one with the smallest change.

In setting up decision tree problems the decision to take no action must be included as an alternative. The event following such a branch has a probability of one. As an example, if a predetermined amount of money is invested and no action is taken at a subsequent decision point, the certain return is a loss of the fixed investment. No sensitivity analysis is performed for branches having a probability of one; it is assumed that such an event is a certainty.

An example will be used to clarify the discussion. Consider the decision tree shown in FIG. 17. The branches selected as having the largest value are A—$B_2$ for the first stage and $C_3$—$D_5$ for the third stage. The most sensitive second stage probability is 0.24 since it is multiplied by the largest value when computing expected value. The expected value at node $B_2$ is:

$$0.41 \times (-10) + 0.35 \times 22.9 + 0.24 \times 77.5 = 22.5^{(1)}$$

The sum of probabilities at a node must always equal one. Consequently, if the most sensitive probability is reduced by $\Delta P_B$ this reduction must be added to one of the other probabilities. Adding it to the probability associated with the smallest value (−10) will produce the minimum increase in expected value for the node. In this case, since the smallest value is negative, adding $\Delta P_B$ to the associated probability also decreases the expected value. The change in the most sensitive probability that will cause the expected value of branch A—$B_2$ to equal that of branch A—$B_1$ may be determined from the following equation.

$$(.41 + \Delta P_B) \times (-10) + 0.35 \times 22.9 + (0.24 - \Delta P_B) \times 77.5 = 20.^{(2)}$$

Subtracting (2) from (1) gives a form which is independent of the individual probabilities:

$$-\Delta P_B \times (-10) + \Delta P_B \times 77.5 = 22.5 - 20.,$$

so

$$\Delta P_B = \frac{22.5 - 20}{77.5 - (-10)} = \frac{\Delta \text{VALUE}}{V_{Max} - V_{Min}} = .028.$$

In other words, if the probability of branch $B_2$—$C_3$ drops by 0.028 and the

probability of branch $B_2 C_1$ increases by the same amount, the expected value of branch A—$B_2$ equals that of branch A—$B_1$.

There are two decision changes that occur when the terminal stage probabilities decrease sufficiently. A reduction in probability can cause a shift in third-stage decisions and in first-stage decisions. The change necessary to cause two third-stage decision branches to have the same expected value is computed using the above formula where now:

$\Delta$ VALUE = difference between third stage branch values,
$V_{Max}$ = maximum value associated with the terminal node being considered.
$V_{Min}$ = minimum value associated with the same node.

The method for computing the change in terminal probability that will cause the value of the first stage decision branch having the largest value to be reduced to the point where it is equal to the next largest first-stage branch value will now be described. Introducing the terminal stage probabilities for node $D_3$ into (1) yields:

$$0.41 \times (-10) + 0.35\,[0.43 \times (-80) + 0.34\,(40) + 0.23 \times (190)] + 0.24 \times 77.5 = 22.5. \quad (3)$$

Introducing the unknown change in terminal probability $\Delta P_D$ gives:

$$0.41 \times (-10) + 0.35\,[(0.43 + \Delta P_D) \times (-80) + 0.34\,(40) + (0.23 - \Delta P_D) \times (190)] + 0.24 \times 77.5 = 20. \quad (4)$$

Subtracting (4) from (3) and rearranging produces:

$$\Delta P_D = \frac{22.5 - 20}{.35\,[190 - (-80)]} = .026$$

Two probability sensitivities are computed for each terminal node which has more than one branch and is in the second section, i.e., connected to node $B_2$. The minimum probability change is selected from the computed values, including $\Delta P_B$, and reported to the decision maker on the display.

The ratio of

$$\frac{\Delta \text{ PROJECT VALUE}}{\Delta P}$$

is also computed for each change in probability and the maximum ratio is the probability most sensitive in changing values.

When the probability most sensitive in changing decision is different from the probability most sensitive in changing values, both are given in a message on the display. The change in probability and the associated branch numbers are also included. In addition, for easy identification these branches are widened in the complete tree which is displayed. FIG. 18 shows the results of an automated decision analysis performed on expected values.

The sensitivity analysis method just described is limited to determining the effects of changes in probability close to the value the decision maker has specified. If the decision maker suspects that he may have made larger errors in assigning some probabilities, especially those not on the decision path, he will want to determine the effects of changes in these probabilities. He can do this by selecting "vary prob" with the light pen. The routines which were previously available for correcting during the input phase then allow him to select any probability he wishes, using the light pen, and he can change it to any desired value. A checking routine insures that he maintains the sum of the probabilities equal to one at the node where he is making changes. A decision maker would also use this type of sensitivity analysis to determine if he should reject an undertaking because a probability could not be determined with sufficient accuracy to insure an adequate return.

The essential features of this sensitivity analysis are to tell the decision maker what his most sensitive probabilities are, how the value of his project changes if he has made an error in their assessment and to give him complete flexibility in determining the effects of large changes in probability. He can observe the effects of changes immediately after making the change, which greatly facilitates his ability

to obtain an understanding of the influence probability has in his decision-making.

12. Sample Problem

The problem used for discussing the operation of the tool which has been developed is that of an oil wildcatter taken from Raiffa (10). The wildcatter must make two decisions:

(1) whether or not to take seismic soundings which will help determine the underlying geological structure at the site and

(2) whether or not to drill.

He assumes the cost of drilling to be $70,000. He also assumes that one of three conditions will exist after drilling. He will have a dryhole which gives him a net loss of the cost of drilling, a wet hole will give a return of $120,000 for a net of $50,000 and a soaking hole will produce $270,000 for a net of $200,000. His best estimate of the probability for the three states is 0.5, 0.3 and 0.1, respectively.

The cost of a seismic test is $10,000 and he allocates three possibilities for the results: no structure, open structure or closed structure with probabilities of 0.41, 0.35 and 0.24, respectively. He estimates that finding an open structure indicating a poor prospect of finding oil, will change the probabilities of his three drilling returns to 0.731, 0.219 and 0.048. If the seismic test shows an open structure his probabilities for a gain improve somewhat to 0.428, 0.342 and 0.288. A closed structure substantially improves the probabilities to 0.208, 0.375 and 0.416.

The data of this project have been used in some of the previous displays. FIG. 10 shows this data displayed on the tree. The top section is for no seismic test. A probability of one is assigned to the first branch of the second stage to indicate this condition. The first three terminal branches are labeled with the probabilities and net returns when no seismic test is performed and a hole is drilled. The next branch represents the situation where nothing is done. The second section represents the condition where a seismic test is performed. The stage two probabilities in the figure are for the aforementioned structure possibilities and lead to the terminal branches having probabilities conditioned by the type of structure. The $10,000 cost for the seismic test is reflected in the terminal values.

The decisions and the expected returns are represented by the branches of stages one and three. The broadened first stage branch (labeled 22.49) and the broadened third stage (labeled 77.50) indicate that the decision maker will receive an expected return of $22,490 by conducting a seismic test and then drilling if both decisions are made at this time. If the drilling decision can be postponed until after the seismic test, the second decision might be different. For instance, if the seismic test showed no structure, the best decision would be not to drill, as indicated by the sixth branch of the third stage where the expected return is given as a loss of $10,000 which is the cost of the seismic test.

The results of performing a sensitivity analysis on expected values are shown in FIG. 18. The branch representing the event having the most sensitive probability is broadened to aid in identification.

At this point, seeing that 11% change in probability materially affects the decision path, the decision maker might wonder how his decisions would be affected if he has been too conservative in his estimate of the probability of obtaining maximum return without a seismic test. If he increases his estimate of this probability by 0.03 to .23 and decreases his estimate of the probability of a dryhole, the computer will change the display to that shown in FIG. 19.

Here the decision maker sees a complete change in both expected value and decision path. Of course, the decision maker probably should not increase the probability of obtaining a maximum return without a seismic test unless he makes an increase in the probability of the same return with a seismic test. This he may do by selecting "vary probability" again. The purpose of this discussion is not to make the decision but to show the flexibility of the tool and how it may be used.

The computation time required to change the probabilities for the no seismic test case was only a few seconds. Thus, in a few minutes the decision maker may explore the effects of changing the probabilities for a number of events. The printer records the changes in probabilities, the corresponding branch numbers, the new decision path and the new expectation for future reference.

After the decision maker has determined his preference function as described in the input section, the computer will repeat the decision tree analysis basing its selections at decision nodes on maximizing expected utility. A display of the results using the preference curve in FIG. 16 is shown in FIG. 20.

The decision path based on expected utility has not changed but the utility of profits as determined by the preference curve, decreases sufficiently with increas-

ing profits as to make the project much less attractive. In addition, drilling is profitable only if a seismic test is performed and a closed structure is obtained.

As with the expected value analysis, the decision maker can select an automated sensitivity analysis based on expected utility. He can also change any of the probabilities and repeat the preference analysis. When he repeats the preference analysis he can by-pass the curve assessment procedure and use the previously assessed curve. It will thus be seen that a tool has been developed which enables a decision maker to make decisions using decision tree analysis with selection of alternates by the computer based on maximization of either expected value or expected utility. A graphic display provides him with patterns showing the interrelationships of data and allows him to select alternate functions. He is also provided with the necessary facilities to determine how closely he must estimate the probability of an event and what his decisions must be in order to achieve the expectation.

The decision maker is provided with sufficient flexibility in changing probabilities, values and his preference curve to give him control over the decision. He is provided with the framework which ties together these fundamental quantities, which in turn are tied closely to real world situations. The computer is sufficiently fast so that he can change one after another and observe the results nearly instantaneously and not lose his train of thought.

The capability can be extended to situations involving more than two decisions by cascading. The output of one or more trees together with other data can be used as the terminal values for a second. Depending on the situation, the most serious limitations may come from the cascading of probabilities. If the decisions cover a period of years, the time value of money should be recognized. In many cases this can be done by adjusting the values when the trees are cascaded.

## BIBLIOGRAPHY

1. Brown, Rex V., "Do Managers Find Decision Trees Useful?" *Harvard Business Review*, Vol. 48, No. 3 (May-June, 1970) pp. 78-79.

2. Fulton, Maurice, "New Factors in Plant Location", *Harvard Business Review*, Vol. 49, No.3 (May-June, 1971), pp 4-6, 8-10, 14-17, 166-168.

3. Hertz, D.B., "Risk Analysis in Capital Investment", *Harvard Business Review*, Vol. 42, No. 1 (Jan.-Feb., 1964), pp. 95-106.

4. Hespos, Richard F., and Strassman, Paul A., "Stochastic Decision Trees for the Analysis of Investment Decisions", *Management Science*, Vol. 11, No. 10 (August, 1965), pp. B-244-259.

5. Private communication, 1969

6. Little, John, D.C. "Models and Managers: The Concepts of a Decision Calculus", *Management Science*, Vol. 16, No. 8, (April, 1970), pp. B-468-485.

7. Miller, George A., "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information", *The Psychological Review*, Vol. 63, No. 2, (March, 1956), pp. 81-97.

8. Miller, Irwin, "Computer Graphics for Decision Making," *Harvard Business Review*, Vol. 47, No. 6 (Nov.-Dec., 1969), pp. 121-132.

9. Pratt, John W., Raiffa, Howard, and Schlaifer, Robert. *Introduction to Statistical Decision Theory*, New York: McGraw-Hill Book Company, 1965.

10. Raiffa, Howard, Decision Analysis: *Introductory Lectures on Choices Under Uncertainty*. Reading, Massachusetts: Addison-Wesley, 1968.

11. Schlaifer, Robert. *Analysis of Decisions Under Uncertainty* New York: McGraw-Hill Book Company, 1969.

12. Schlaifer, Robert. *Manual of Cases on Decision Under Uncertainty with Anaylses and Teaching Notes*. New York: McGraw-Hill Book Company, 1968.

13. Schlaifer, Robert. *"Computer Programs for Managerial Economics"*. Obtainable from Professor Arthur Schlaifer, Jr., Harvard Business School, Boston, Mass. 02163.

14. Simon, Herbert A., *The Shape of Automation*. New York Harper & Row, 1965, pp. 58-61.

15. Sutherland, John "Scientific Management—Too Often Just an Exercise in Over-Simplification", *Computer Decisions*, Vol. 3, No. 1 (January, 1971), p. 60.

16. Swalm, Ralph O. "Utility Theory — Insights into Risk Taking", *Harvard Business Review*, Vol. 44, No. 6 (Nov.-Dec., 1966), pp. 123-136.

Brief Description of the Program Subroutines

The following is a brief description of the program subroutines setting forth their respective functions

| NAME | CALLED BY | FUNCTIONS |
|---|---|---|
| MAIN 2 | | MAIN LINE PROGRAM—retrieves from and stores on disk. Sets scales. Selection of "preference", "sensitivity", "fini", "store", "correct", "vary prob". Prints expectations. |
| ACTS | MAIN 2 | Broadens selected branches, displays and prints expectations. Prints numbers of selected branches. |
| ASPRF | MAIN 2 | Assess and verifies DM's preference curve, has menu of "calculate" and "preference". |
| DTREE | MAIN 2 | Zeros expectations and decisions in arrays. Averages out and folds back from array "D" to "C", "B", and and "A" in column 3. In arrays "A" and "C", places maximum value for each node in column 4. Makes all utiles positive for preference analysis. |
| ICORR | MAIN | Corrects probabilities and terminal values. |
| ICRVP | INITV<br>ICORR | Corrects terminal values and terminal probabilities, checks for sum of probabilities equal to one. |
| INITB | MAIN | Input and correction for stage 2 probabilities. |
| INITV | MAIN | Obtains terminal values and terminal probabilities from DM and displays them on tree, provides skip node function and marker, checks node probability sum and provides for its correction, assigns act flags for stage 3. |
| INPUT | INITV<br>ASPRF<br>ICRVP | Asks DM for input values, checks for legal characters, compacts divides previous values by 1000 when M is used. |
| ISTGB | INITB<br>ICORR | Asks DM for stage 2 probabilities, Checks sum at a node, repeats for correction. Corrects or varies stage 2 probabilities. Assigns and corrects event flags for stage 2, act flags for stage 1. |
| MAIN | MAIN 2 | Obtains input data from DM and provides correction facilities. Zeros arrays for new problem, displays first four branches of DT, displays scale change marker. |
| MENU | MAIN<br>INITV | Displays choices of: section to be magnified, "all sections", "stage 2", "store". |
| PICRD | ASPRF | Displays axes and labels for preference curve. |
| PINVT | MAIN 2 | Transforms expected utiles in column 4 of arrays "A" and "C" back into same dimensions as terminal values. (act/event flag must be set for each branch). |
| PREFR | MAIN 2 | Transforms terminal values in array "D" into preference values. Shifts PI's to make them all positive. |
| PTREE | MAIN<br>TREE | Displays stages 2, 3, 4 of tree, (pruned or unpruned), terminal values or utiles, and terminal probabilities, spreads terminal values. |

| | | |
|---|---|---|
| SNSTY | MAIN 2 | Determines probability most sensitive in changing decision path and probability on decision path most sensitive in changing values. |
| SPRED | SNSTY | Broadens the branch at each node that is associated with the probability having the greatest sensitivity. |
| TREE | MAIN 2<br>MAIN<br>INITB | Writes first four branches (Stage 1) of tree. |
| VINVT | MAIN 2<br>SNSTY | changes input (V) from utiles back to terminal value units. |

The Program Listing

The following is a listing of the instructions of a FORTRAN program employing principles of the present invention.

```
PAGE 1      MAIN2

// JOB                                                         MAIN 2
  LOG DRIVE    CART SPEC    CART AVAIL    PHY DRIVE
    0000         0021         0021          0000

V2 M08   ACTUAL 32K   CONFIG 32K

// FOR
*ONE WORD INTEGERS
*IOCS(TYPEWRITER,PLOTTER,KEYBOARD,CARD,DISK)
*IOCS (1403 PRINTER)
*LIST SOURCE PROGRAM
**MAINLINE PROGRAM R DE LANO—3/71 REV 08/01/71
```

PAGE 2 MAIN2 MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71

```
      DEFINE FILE 4 (99, 320, U, IREC)
      INTEGER PNAME(20,16)
      DIMENSION PRFR(6)
      COMMON D(512),C(4,64), B(4,16),A(4,4)
C PRFR CONTAINS PIS AND PSHFT
C IPRF— PREFERENCE CURVE ALREADY ASSESSED. SET HERE
C IFLG4 SET IN INITV TO SKIP INPUT INSTRUCTIONS AFTER FIRST TIME
C IFLAG SET HERE— MATRICES CONTAIN UTILES
C ISTOR = 0 TO CALCULATE, 1 TO STORE                              MAIN
C IRSME = 1 TO CONTINUE PROBLEM PREVIOUSLY STORED                 HERE
C IVRPB—PROBABLITIES BEING VARIED—PRINT NEW PROBABILITIES. SET HERE.
C GIO BRACKETS (ICODE)
C 1 — 4 SECTIONS, 6 ALL SECTIONS,7 STAGE 2                        MENU
C 8 — 71 TERMINAL NUMBERS (SAME FOR EACH SECTION)                 INITV
C 72— FINI                                                        HERE
C 73 — 88 STAGE 2 PROB                                            INITB
C 89 SKIP MARKER                                                  INITV
C 90 CALCULATE                                                    MAIN, ASPRF
C 100—PREFERENCE                                                  HERE, ASPRF
C 101— SENSITIVITY                                                HERE
C 102 — 117 DECISION TREE PROBLEMS STORED                         HERE
C 118 — NEW                                                       HERE
C 119 — STORE                                                     HERE
C 120 — CORRECT, VARY PROB                                        HERE, MAIN
C 121 — 126 PREFERENCE CURVE                                      ASPRF
C 127 — ASSESS                                                    HERE
C DISK RECORD NO. 1 CONTAINS NAMES OF PROBLEMS STORED
C RECORDS 2 — 7 PROB 1 DATA, 8 — 13 PROB 2 DATA, ETC FOR 16 PROBLEMS
      IKEYB = 6
      IPLT = 7
      IDSPL = 1
      IPRNT = 5
      IMFLG = 0
      IFLG4 = 0
      IRSME = 0
      ISTOR = 0
      ICRCT = 0
```

```
PAGE 3      MAIN2      MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71
            IVRPB = 0
            IPRF = 0
            IFLAG = 0
            CALL DATSW(0,ISW)
C DISABLE 2250 CANCEL KEY
            CALL IREQ(KREQ)
            READ(4'1)((PNAME(I,J),I = 1,20),J = 1,16)
            WRITE(IDSPL,1058)
C
C SET UP MENU OF PROBLEMS
C
            XSCLE = 1.
            YSCLE = .9
            YDLT = .36/YSCLE
            XPOST = 0.
            YPOST = 0.
            YPOS = 9.72/YSCLE+YPOST
            YPOSN = YPOS
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
            DO 5 I = 102,117
            CALL GIOLB(I)
            YPOSN = YPOSN—YDLT
            CALL FPLOT(1,0.,YPOSN)
            II = I—101
            WRITE(IPLT,1059)II,(PNAME(K,II),K = 1,20)
5           CONTINUE
            CALL GIOLB(118)
            YPOSN = YPOSN—YDLT
            CALL FPLOT(1,0.,YPOSN)
            WRITE(IPLT,1060)
            CALL GIORB
C
C LCOP ON GIOLP UNTIL SELECTION FROM PROBLEM LIST OR 'NEW'
C
10          CALL GIOLP(ICODE,IX,IY)
            IF(ICODE)10,10,15
15          IF(CODE—118)17,21,21
```

```
PAGE 4      MAIN2      MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71

C HERE FOR OLD PROBLEM
17          IRSME = 1
C READ SELECTED PROBLEM DATA FROM DISK
            ISTG2 = 1
            IREC = 2+(ICODE—102)*6
            READ(4'IREC)A,B,C,D,PRFR
            IFLAG = B(1,3)
C HERE TO DISPLAY OLD PROBLEM, START NEW, CORRECT OR VARY PROB
21          XSCLE = 1.
            YSCLE = .9
            YDLT = .36/YSCLE
            XPOST = 0.
            YPOST = 0.
            YPOS = 9.72/YSCLE+YPOST
C POSITION BEAM OFF SCREEN TO ERASE
            CALL FPLOT(1.,1100.,0.)
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
            YPOSN = YPOS
            CALL   MAIN(MAXD,MIND,IRSME,ISTG2,ISTOR,ICRCT,IFLG4,IVRPB,
           *IMFLG)
            IF(ICRCT)23,23,210
210         WRITE(IDSPL,1066)
            GO TO 21
C HERE AFTER STORE OR CALCULATE SELECTED
23          XSCLE = 1.
            YSCLE = .9
            YDLT = .36/YSCLE
            XPOST = 0.
            YPOST = 0.
            YPOS = 9.72/YSCLE+YPOST
C POSITION BEAM OFF SCREEN TO ERASE
            CALL FPLOT(1.,1100.,0.)
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
            YPOSN = YPOS
            IF(ISTOR)40,40,24
C HERE TO PROVIDE SELECTION OF RECORD NAME FOR STORAGE OF PROBLEM ON DISK
```

```
PAGE 5      MAIN2      MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71
24          WRITE(IDSPL,1061)
            DO 25 I = 102,117
            CALL GIOLB(I)
            YPOSN = YPOSN—YDLT
            CALL FPLOT(1,0.,YPCSN)
            II = I—101
            WRITE(IPLT,1059)II,(PNAME(K,II),K = 1,20)
25          CONTINUE
            CALL GIORB
C
C LOOP ON GIOLP UNTIL SELECTION
C
30          CALL GIOLP(ICODE,IX,IY)
            IF(ICODE)30,30,32
32          IREC = 2+(ICODE—102)*6
C WRITE PROBLEM DATA INTO SELECTED RECORD OF DISK
C STORE IMFLG IN B(1,2), IFLAG IN B(1,3)
            B(1,2) = IMFLG
            B(1,3) = IFLAG
            WRITE(4'IREC)A,B,C,D,PRFR
            KREC = ICODE—101
            WRITE(IDSPL,1063)
            READ(IKEYB,1062)(PNAME(J,KREC),J = 1,20)
            WRITE(4'1)PNAME
            YPOSN = YPOS
            GO TO 65
C COMPUTE PROPOSITION VALUE AND DISPLAY DECISIONS
40          BIT = 55.
            YTOP = 11.
            ISTG2 = 1
            ISTG4 = 1
            ICRCT = 0
C POSITION BEAM OFF SCREEN TO ERASE
            CALL FPLOT(1.,1100.,0.)
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
            IF(IFLAG)402,402,59
402         CALL      DTREE(IFLAG,VALUE)
```

PAGE 6 MAIN2 MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71

```
          ICRCT=0
          CALL    TREE(ISTG2,ISTG4,YTOP,BIT,ICRCT)
          CALL    ACTS(YTOP,IAMAX)
          WRITE(IPRNT,1088)VALUE
          WRITE(IDSPL,1057)
C
C MENU SELECTION FROM— PREFERENCE, SENSITIVITY, STORE, CORRECT, FINI
C
41        YPOSN=YPOS
          CALL GIOLB(100)
          CALL FPLOT(1,0.,YPOSN)
          WRITE(IPLT,1054)
          YPOSN=YPOSN—YDLT
          CALL GIOLB(101)
          CALL FPLOT(1,0.,YPOSN)
          WRITE(IPLT,1055)
          CALL GIOLB(119)
          YPOSN=YPOSN—YDLT
          CALL FPLOT(1,0.,YPOSN)
          WRITE(IPLT,1064)
          CALL GIOLB(120)
          YPOSN=YPOSN—YDLT
          CALL FPLOT(1,0.,YPOSN)
          WRITE(IPLT,1065)
          CALL GIOLB(72)
          YPOSN=YPOSN—YDLT
          CALL FPLOT(1,0.,YPOSN)
          WRITE(IPLT,1048)
          CALL GIORB
          YPOSN=YPOS
50        CALL GIOLP(ICODE,IX,IY)
C
C LOOP ON GIOLP UNTIL SELECTION
C
          IF(ICODE)50,50,51
51        IF(ICODE—101)57,52,53
```

```
PAGE 7     MAIN2     MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71
52         CALL     SNSTY(IAMAX,YTOP,BIT,IFLAG,PI0,PI25,PI50,PI75,PI100,
          *PSHFT)
           CALL GIOLB(120)
           YPOSN = YPOS—5*YDLT
           CALL FPLOT(1,0.,YPOSN)
           WRITE(IPLT,1076)
           GO TO 41
53         IF(ICODE—119)57,531,54
531        ISTOR = 1
           GO TO 23
54         ICRCT = 1
           WRITE(IDSPL,1092)
           IVRPB = 1
           GO TO 21
57         IF(ICODE—100)65,59,65
C HERE FOR PREFERENCE ANALYSIS
C POSITION BEAM OFF SCREEN TO ERASE
59         CALL FPLOT(1.,1100.,1100.)
           CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
           IMFLG = B(1,2)
C SKIP ASSESSMENT AT DM'S OPTION IF ALREADY DONE
           IF(IFLAG)596,596,592
592        WRITE(IDSPL,1090)
C MENU OF PREFERENCE AND ASSESS
           YPOSN = YPOS
           CALL GIOLB(100)
           CALL FPLOT(1,0.,YPCSN)
           WRITE(IPLT,1054)
           YPOSN = YPOS—6*YDLT
           CALL GIOLB(127)
           CALL FPLOT(1,0.,YPOSN)
           WRITE(IPLT,1091)
           CALL GIORB
           YPOSN = YPOS
           CALL FPLOT(1,0.,0.)
593        CALL GIOLP(ICODE,IX,IY)
```

```
PAGE 8      MAIN2      MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71
C
C LOOP ON GIOLP UNTIL SELECTION
C
            IF(ICODE)593,593,594
 594        PI0 = PRFR(1)
            PI25 = PRFR(2)
            PI50 = PRFR(3)
            PI75 = PRFR(4)
            PI100 = PRFR(5)
            PSHFT = PRFR(6)
            CALL GIODE (100)
            CALL GIODE (127)
            DO 5952 N = 1,256
            IF(D(N+256))5952,5952,5951
 5951       V = D(N)
            CALL     VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
            D(N) = V
 5952       CONTINUE
            IF(ICODE—100)598,598,596
C ASSESS PREFERENCE CURVE
 596        CALL     ASPRF(PI0,PI25,PI50,PI75,PI100,IMFLG,DLRS)
            IPRF = 1
 598        CALL     PREFR(PI0,PI25,PI50,PI75,PI100,PSHFT)
            PRFR(1) = PI0+PSHFT
            PRFR(2) = PI25+PSHFT
            PRFR(3) = PI50+PSHFT
            PRFR(4) = PI75+PSHFT
            PRFR(5) = PI100+PSHFT
            PRFR(6) = PSHFT
C SET FLAG TO INDICATE UTILES ARE IN ARRAYS
            IFLAG = 1
C HERE TO COMPUTE PREFERENCE FOR OLD OR NEW PROJECT
            ICRCT = 0
            CALL     DTREE(IFLAG,VALUE)
            CALL     PINVT(PI0,PI25,PI50,PI75,PI100,PSHFT)
            CALL     TREE(ISTG2,ISTG4,YTOP,BIT,ICRCT)
```

PAGE 9 MAIN2 MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71

```
      CALL      ACTS(YTOP,IAMAX)
      WRITE(IPRNT,1015)A(4,IAMAX)
      WRITE(IDSPL,1015)A(4,IAMAX)
      WRITE(IDSPL,1089)
      CALL GIOLB(120)
      YPOSN=YPOS—5*YDLT
      CALL FPLOT(1,0.,YPCSN)
      WRITE(IPLT,1076)
      GO TO 41
1015  FORMAT (///' THE PREFERENCE VALUE OF YOUR PROJECT IS',F10.3,/)
1048  FORMAT('FINI')
1054  FORMAT('PREFERENCE')
1055  FORMAT('SENSITIVITY')
1057  FORMAT('I HAVE COMPUTED THE VALUE OF YOUR PROPOSITION'/'THE MOST V
     *ALUABLE CHOICES ARE PRINTED AND SELECTED BRANCHES ARE HEAVY'/'SELE
     *CT'PREFERENCE TO DO A UTILITY ANALYSIS'/ 'SENSITIVITY FOR SENSITIV
     *ITY ANALYSIS, FINI WHEN FINISHED, PLEASE SELECT')
1058  FORMAT('SELECT DESIRED PROBLEM OR "NEW" WITH LIGHT PEN'/'WAITING
     * FOR YOUR SELECTION'//)
1059  FORMAT(3X,I2,'..',20A2)
1060  FORMAT(5X,'..NEW')
1061  FORMAT('YOU WISH TO STORE YOUR PROBLEM'/'I WILL DISPLAY THE LIST O
     *F STORED PROBLEMS'/'SELECT A SPACE WITH THE LIGHT PEN'/)
1062  FORMAT(20A2)
1063  FORMAT('TYPE THE NAME TO BE ASSIGNED BELOW, 20 CHARACTERS MAX.'/
     *'CHECK WHAT YOU HAVE TYPED'/
     *'THEN PUSH BOTH ALTN CODING AND END AT THE SAME TIME')
1064  FORMAT('STORE')
1065  FORMAT('CORRECT')
1066  FORMAT('YOU WISH TO CHANGE DATA'
     */'SELECT A SECTION TO CHANGE TERMINAL DATA'/
     *'SELECT "ALL SECTIONS" TO CHANGE SECOND STAGE PROB')
1076  FORMAT('VARY PROB')
1088  FORMAT(' THE EXPECTED VALUE OF YOUR PROJECT IS',F10.3)
1089  FORMAT('WAITING FOR YOUR SELECTION FROM THE MENU'/)
1090  FORMAT('I HAVE YOUR PREFERENCE CURVE.'/
```

```
PAGE 10      MAIN 2      MAINLINE PROGRAM R DE LANO — 3/71 REV 08/01/71

             *'SELECT "PREFERENCE" TO USE IT'/
             *'SELECT "ASSESS" TO PROVIDE A NEW CURVE'/
             *'WAITING FOR YOUR SELECTION')
      1091    FORMAT('ASSESS')
      1092    FORMAT('YOU WISH TO CHANGE DATA'/
             *'SELECT A SECTION TO CHANGE TERMINAL DATA. REMEMBER VALUES ERASED'
             */'SELECT"ALL SECTIONS" TO CHANGE SECOND STAGE PROB.')
      65      CALL EXIT
              END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON  1696  VARIABLES  412  PROGRAM  1820

END OF COMPILATION

// DUP

*DELETE                MAIN2
CART ID 0021      DB ADDR 2C54      DB CNT 0076

*STORE        WS UA  MAIN2
CART ID 0021      DB ADDR 3490      DB CNT 0076


PAGE 1      ACTS

// JOB                                            ACTS

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021        0021         0000

V2 M08      ACTUAL 32K      CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**DISPLAY DECISIONS R DE LANO 5/70 REV 3/20/71
```

```
PAGE 2    ACTS    DISPLAY DECISIONS R DE LANO 5/70 REV 3/20/71

          SUBROUTINE ACTS(YTOP,IAMAX)
          COMMON D(512),C(4,64), B(4,16),A(4,4)
C PRINTS DECISIONS TAKEN. BROADENS BRANCHES AND PLACES VALUES ON THEM
C DISPLAYS DECISIONS TAKEN.
C CALLED BY MAIN2
C IAMAX IS NO. OF NODE HAVING LARGEST VALUE — COMPUTED HERE
          IPRNT = 5
          IPLT = 7
          WRITE(IPRNT,1014)
          DO 4 N = 1,64
          IF(C(4,N)—0.)2,4,2
C HERE TO BROADEN SELECTED BRANCHES AND WRITE VALUES— 'C' BRANCHES
2         WRITE(IPRNT,1016)N
          IB = N/4+1
          YBB = YTOP—IB*.64+.42
          DO 3 ISPRD = 1,4
          YCC = YTOP—N*.16+.182—ISPRD*.015
          CALL FPLOT(1,4.1,YBB)
          CALL FPLOT(2,5.95,YCC)
3         CONTINUE
          YBBB = YBB+.2
          CALL FCHAR(3.7,YBBB,.1,.1,0.)
          WRITE(IPLT,1051)C(4,N)
4         CONTINUE
          DO 8 N = 1,16
          IF(B(4,N)—0.)6,8,6
6         WRITE(IPRNT,1018)N
8         CONTINUE
          DO 12 N = 1,4
          IF(A(4,N)—0.)10,12,10
C HERE TO BROADEN SELECTED BRANCHES AND WRITE VALUES— 'A' BRANCHES
10        WRITE(IPRNT,1020)N
          IAMAX = N
          HLFY = YTOP/2
          YA = YTOP—N*2.56+1.28
          DO 11 ISPRD = 1,4
          YAA = YA+.03—ISPRD*.02
```

```
PAGE 3      ACTS      DISPLAY DECISIONS R DE LANO 5/70 REV 3/20/71

            CALL FPLOT(1,.1,HLFY)
            CALL FPLOT(2,1.95,YAA)
   11       CONTINUE
            YAAA = YTOP—N*1.28—2.56
            XA = 0.
            IF(N—2)15,14,13
   13       IF(N—3)14,14,15
   14       XA = .25
   15       CALL FCHAR(XA,YAAA,.1,.1,0)
            WRITE(IPLT,1051)A(4,N)
   12       CONTINUE
 1014       FORMAT(//,2X,'THE FOLLOWING DECISION BRANCHES WERE SELECTED')
 1016       FORMAT(//,2X,'C—',I3)
 1018       FORMAT(//,2X,'B—',I3)
 1020       FORMAT(//,2X,'A—',I3)
 1051       FORMAT(F10.2)
            RETURN
            END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR ACTS
 COMMON  1696  VARIABLES  24  PROGRAM  410

RELATIVE ENTRY POINT ADDRESS IS 0076 (HEX)

END OF COMPILATION

// DUP

*DELETE              ACTS
CART ID 0021     DB ADDR 2C54     DB CNT 001C

*STORE       WS  UA  ACTS
CART ID 0021     DB ADDR 34F6     DB CNT 001C
```

```
PAGE 1     ASPRF

// JOB                                          ASPRF

  LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
    0000        0021        0021         0000

V2 M08    ACTUAL 32K    CONFIG 32K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
** ASSESS PREFERENCE CURVE R DE LANO 5/9/71
```

```
PAGE 2      ASPRF      ASSESS PREFERENCE CURVE R DE LANO 5/9/71

            SUBROUTINE ASPRF(PI0,PI25,PI50,PI75,PI100,IMFLG,DLRS)
C CALLED BY MAIN 2
C ASPRF ASSESS 3 POINTS ON PREFERENCE CURVE, DISPLAYS CURVE,
C AND VERIFIES WITH TWO ADDITIONAL LOTTERIES, PI25—PI100, PI25—PI75.
C IPFLG INDICATES TO INPUT THIS SUBROUTINE CALLED IT. SET HERE
            COMMON D(512),C(4,64), B(4,16),A(4,4)
            IPFLG = 1
            IDSPL = 1
            IPLT = 7
            IPRNT = 5
            CALL SCALF(1.,.9,0.,—2)
C SET       ID = 1 FOR ERROR ROUTINE IN INPUT
            ID = 1
C DETERMINE LARGEST AND SMALLEST TERMINAL VALUES
            TVMAX = D(1)
            TVMIN = TVMAX
            DO 10 I = 1,256
            IF(TVMAX—D(I))5,7,7
  5         TVMAX = D(I)
  7         IF(TVMIN—D(I))10,10,9
  9         TVMIN = D(I)
 10         CONTINUE
            PI0 = TVMIN
            PI100 = TVMAX
            CALL      PICRD(PI0,PI100)
 11         WRITE(IDSPL,1078)PI100,PI0
            IF(IMFLG)13,13,15
 13         WRITE(IDSPL,1079)
 15         CALL      INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
            PI50 = DLRS
            WRITE(IDSPL,1078)PI50,PI0
            CALL      INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
            PI25 = DLRS
            WRITE(IDSPL,1078)PI100,PI50
            CALL      INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
            PI75 = DLRS
C PLOT PREFERENCE CURVE
```

```
PAGE 3      ASPRF      ASSESS PREFERENCE CURVE R DE LANO 5/9/71
            CALL GIOLB(121)
            CALL FPLOT(1,2.,.9)
            XI25=2.+4*(PI25—PI0)/(PI100—PI0)
            CALL FPLOT(2,XI25,2.15)
            CALL GIOLB(122)
            XI50=2.+4*(PI50—PI0)/(PI100—PI0)
            CALL FPLOT(2,XI50,3.4)
            CALL GIOLB(123)
            XI75=2.+4*(PI75—PI0)/(PI100—PI0)
            CALL FPLOT(2,XI75,4.65)
            CALL GIOLB(124)
            CALL FPLOT(2,6.0,5.9)
            CALL GIORB
            CALL FCHAR(1.,7.,.2,.2,0.)
            WRITE (IPLT,1084)
            CALL FCHAR(1.,6.7,.1,.1,0.)
            WRITE(IPLT,1087)
C HERE TO ADD TWO CHECK POINTS
            WRITE(IDSPL,1081)
            WRITE(IDSPL,1082)PI25,PI100
            CALL      INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
            PI625=DLRS
            WRITE(IDSPL,1082)PI25,PI75
            CALL      INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
            PI50C=DLRS
C PLOT THE NEW VALUES
            CALL GIOLB(125)
            XI625=2.+4*(PI625—PI0)/(PI100—PI0)
            CALL FPLOT(1,XI625,4.03)
            CALL FPLOT(—2,XI625,4.03)
            CALL POINT(1)
            CALL FPLOT(1,XI625,4.03)
            CALL GIOLB(126)
            XI50C=2+4*(PI50C—PI0)/(PI100—PI0)
            CALL FPLOT(—2,XI50C,3.4)
            CALL POINT(1)
            CALL GIORB
```

PAGE 4 ASPRF ASSESS PREFERENCE CURVE R DE LANO 5/9/71

```
      WRITE(IDSPL,1083)
C
C MENU— CALCULATE, PREFERENCE
C
      CALL GIOLB(90)
      CALL FPLOT(1,0.,8.0)
      WRITE(IPLT,1046)
      CALL GIOLB(100)
      CALL FPLOT(1,0.,7.6)
      WRITE(IPLT,1054)
      CALL GIORB
C LCOP ON GIOLP UNTIL SELECTION
20    CALL GIOLP(ICODE,IX,IY)
      IF(ICODE—90)20,50,29
29    IF(ICODE —100)20,31,20
31    DO 40 I = 121,126
      CALL GIODE(I)
40    CONTINUE
      GO TO 11
C ERASE SCREEN AND RESET SCALE
50    CALL FPLOT(1,1000.,0.)
      CALL SCALF(1.,.9,0.,+2.0)
      RETURN
1046  FORMAT('CALCULATE')
1054  FORMAT('PREFERENCE')
1078  FORMAT('TYPE BELOW YOUR CME FOR THE 50—50 LOTTERY'/
     *'HAVING A MAX GAIN OF',F10.3,' AND A MIN GAIN/LOSS OF',F10.3)
1079  FORMAT('YOU DID NOT USE M IN INPUT, PLEASE DO NOT USE IT HERE')
1081  FORMAT('I WOULD LIKE TO BE SURE YOUR CURVE REFLECTS YOUR ATTITUDES
     *'/'LETS TRY 2 ADDITIONAL LOTTERIES')
1082  FORMAT('TYPE BELOW YOUR CME FOR THE 50—50 LOTTERY',F10.3,',',F10.3
     *)
1083  FORMAT('DO THE TWO NEW VALUES LIE CLOSE TO THE CURVE'/
     *'IF THEY DO,SELECT"CALCULATE" TO OBTAIN PREFERENCE VALUE'/
     *'OTHERWISE SELECT"PREFERENCE" TO REDO THE CURVE'/
     *'WAITING FOR YOUR SELECTION')
```

```
PAGE 5      ASPRF       ASSESS PREFERENCE CURVE R DE LANO 5/9/71

1087        FORMAT('COMPARE IT TO THE STRAIGHT LINE FOR AN EMV''ER')
1084        FORMAT('THIS IS YOUR PREFERENCE CURVE')
            END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR ASPRF
 COMMON  1696  VARIABLES  34   PROGRAM   940

RELATIVE ENTRY POINT ADDRESS IS 0183 (HEX)

END OF COMPILATION

// DUP

*DELETE                  ASPRF
CART ID 0021      DB ADDR 2C54       DB CNT 003D

*STORE        WS  UA  ASPRF
CART ID 0021      DB ADDR 34D2       DB CNT 003D



PAGE 1      DTREE                                        DTREE

// JOB

  LOG DRIVE    CART SPEC   CART AVAIL   PHY DRIVE
    0000         0021        0021         0000

V2 M08      ACTUAL 32K     CONFIG  32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**FOUR STAGES FOUR BRANCHES PER STAGE-- R DE LANO 5/70 REV 08/02/71
```

```
PAGE 2      DTREE      FOUR STAGES FOUR BRANCHES PER STAGE— R DE LANO 5/70
            REV 08/02/71

            SUBROUTINE DTREE(IFLAG,VALUE)
C CALLED TWICE BY MAIN
C ZEROS EXPECTATIONS AND DECISIONS IN ARRAYS
C AVERAGES OUT AND FOLDS BACK FROM 'D' TO 'C', 'B', 'A'
C IFLAG SET IN MAIN2 WHEN COMPUTING PREFERENCE VALUES
C EACH NODE HAS UP TO FOUR BRANCHES
C ARRAY D CONTAINS 1)(1TO256)TERMINAL VALUES AND 2) THEIR PROBALBITIES (257TO512)
C ARRAY C CONTAINS 1)ACT/EVENT FLAG,2) DISTRIBUTION FUNCTION NUMBER OR
C PROBABLITY, 3) EXPECTED VALUE FOLDED BACK FROM TERMINALS AND 4)LARGEST VALUE
C ASSOCIATED WITH EACH NODE
C ACT/EVENT FLAG = 2/1.EACH EXISTING ACT BRANCH MUST BE CODED '2'. SET IN ISTGB
C ARRAYS B AND A ARE SIMILAR TO C, NODES IN B FLOW TO C, AND A TO B.
            REAL NEXT
            COMMON D(512),C(4,64), B(4,16),A(4,4)
            IPRNT = 5
            IDSPL = 7
            VALUE = 0.
            DO 4 J = 1,64
C ZERO PREVIOUS EXPECTATIONS AND DECISIONS IN ARRAY 'C'
            C(3,J) = 0.
            C(4,J) = 0.
            DO 3 N = 1,4
            JN = 4*(J—1)+N
            C(3,J) = C(3,J)+D(JN)*D(JN+256)
3           CONTINUE
C PI VALUES IN ARRAY C MUST BE POSITIVE FOR PREFERENCE ANALYSIS
            IF(IFLAG)4,4,300
300         E = C(3,J)
            C(3,J) = ABS(E)
4           CONTINUE
C ARRAY C(3,1TO64) NOW CONTAINS EXPECTED VALUES FOLDED BACK FROM TERMINALS
            DO 18 I = 1,16
C ZERO PREVIOUS EXPECTATIONS IN ARRAY B
            B(3,I) = 0.
            B(4,I) = 0.
            II = 4*I—3
            IF(C(1,II)—1.)18,49,12
```

```
PAGE 3    DTREE     FOUR STAGES FOUR BRANCHES PER STAGE— R DE LANO 5/70
REV 08/02/71

C HERE IF NODE C(1,II) IS AN ACT.SELECT MAXIMUM. PLACE IN ARRAY B.
12        RMAX = C(3,II)
          C(4,II) = RMAX
          DO 16 K = 2,4
C TEST FOR ACTIVE BRANCH.
          IK = 4*(I—I)+K
          IF(C(1,IK))16,16,13
13        NEXT = C(3,IK)
          IF(NEXT—RMAX)16,16,14
14        RMAX = NEXT
C PLACE LARGEST VALUE FOR NODE IN COLUMN 4 OF C AND COLUMN 3 OF B
          C(4,II) = 0.
          C(4,IK) = NEXT
16        CONTINUE
          B(3,I) = RMAX
18        CONTINUE
C ARRAY B(3,1TO16) NOW CONTAINS EXPECTED VALUES FOLDED BACK FROM ARRAY C.
C COLUMN 4 OF ARRAY C INDICATES WHICH ACT NODE WAS LARGEST.
          DO 30 I = 1,4
C ZERO PREVIOUS EXPECTATIONS AND DECISIONS IN ARRAY 'A'
          A(3,I) = 0.
          A(4,I) = 0.
          II = 4*I—3
          IF(B(1,II)—1.)30,20,49
C HERE IF NODE B(1,II) IS AN EVENT. PLACE EXPECTED VALUE IN ARRAY A.
20        DO 22 L = 1,4
          IL = 4*(I—1)+L
          A(3,I) = A(3,I)+B(2,IL)*B(3,IL)
22        CONTINUE
30        CONTINUE
          IF(A(1,1)—1.)49,49,36
36        RMAX = A(3,1)
          A(4,1) = RMAX
          DO 40 K = 2,4
C TEST FOR ACTIVE BRANCH.
          IF(A(1,K)—0.)40,40,37
37        NEXT = A(3,K)
```

```
PAGE 4      DTREE      FOUR STAGES FOUR BRANCHES PER STAGE— R DE LANO 5/70
REV 08/02/71

            IF(NEXT—RMAX)40,40,38
      38    RMAX = NEXT
            A(4,K—1) = 0.
            A(4,K) = RMAX
      40    CONTINUE
            VALUE = RMAX
            RETURN
      49    WRITE(IPRNT,1013)
            WRITE(IDSPL,1013)
    1013    FORMAT(/' SYSTEM ERROR. EACH ACTIVE BRANCH IN ARRAY "C" MUST BE
           *FLAGGED 2'/
           *' EACH ACTIVE BRANCH IN ARRAY "B" MUST BE FLAGGED 1 OR HAVE A PR
           *OBABILITY'/
           *' EACH ACTIVE BRANCH IN ARRAY "A" MUST BE FLAGGED 2')
            RETURN
            END

features supported
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR DTREE
 COMMON  1696  VARIABLES  20  PROGRAM  596

RELATIVE ENTRY POINT ADDRESS IS 0084 (HEX)

END OF COMPILATION

// DUP

*DELETE              DTREE
CART ID 0021    DB ADDR 2C54    DB CNT 0029

*STORE        WS UA  DTREE
CART ID 0021    DB ADDR 34DF    DB CNT 0029
```

```
PAGE 1     ICORR
                                                  ICORR
// JOB

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021        0021         0000

V2 M08     ACTUAL 32K     CONFIG 32K

// FOR
**CORRECT ALL PROB AND TERM VALUES— R DE LANO 3/27/71 REV 08/01/71
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
```

```
PAGE 2    ICORR    CORRECT ALL PROB AND TERM VALUES— R DE LANO 3/27/71
REV 08/01/71

          SUBROUTINE ICORR(ISECT,YTOP,ICODE,YPOS,YDLT,IMFLG,ICRCT,IVRPB,BIT)
C CALLED TWICE BY MAIN
C CONTAINS FINAL CORRECTION ROUTINES
C ICODE FROM GIOLP. LESS THAN 8—RETURN, 8 TO 71—ERASE NODE VALUE AND PROB,
C 72—ERROR MSG, 73—88 ERASE STAGE 2 PROB, 90—RETURN AND CALCULATE
          COMMON D(512),C(4,64), B(4,16),A(4,4)
          ISAVE = 0
          IKEYB = 6
          IPLT = 7
          IDSPL = 1
          CALL GIOLB(90)
          YPOSN = YPOS—7*YDLT
          CALL FPLOT(1,0.,YPOSN)
          WRITE(IPLT,1046)
          CALL GIORB
C SKIP SELECTION WHEN CHANGING STAGE 2 PROB
          IF(ICODE—73)8,2,2
2         IF(ICODE—88)38,38,33
8         WRITE(IDSPL,1052)
C
C LOOP ON GIOLP UNTIL SELECTION MADE
C
10        CALL GIOLP(ICODE,IX,IY)
          IF(ICODE)10,10,20
20        IF(ICODE—8)30,32,32
30        RETURN
32        IF(ICODE—72)34,33,36
C SYSTEM ABORT
33        WRITE(IDSPL,1023)
          GO TO 10
C HERE TO CORRECT TERMINAL VALUES AND PROB OF SELECTED TERMINAL NODE
34        CALL    ICRVP(ICODE,ISECT,IMFLG,YTOP,IVRP8)
          ISAVE = 1
          GO TO 8
36        IF(ICODE—88)38,38,40
C HERE TO CORRECT STAGE 2 PROB
```

```
PAGE 3     ICORR      CORRECT ALL PROB AND TERM VALUES— R DE LANO 3/27/71
REV 08/01/71

38        ISTRT = (ICODE—69)/4
          IFINI = ISTRT
          DO 39 I = 69,72
          CALL GIODE(I+ISTRT*4)
39        CONTINUE
          IF(ISAVE)391,391,392
391       ISECT = 6
392       CALL    ISTGB(ISTRT,IFINI,YTOP,BIT,ISECT,ICODE,IVRPB)
          WRITE(IDSPL,1053)
          GO TO 10
40        IF(ICODE—90)33,42,42
1023      FORMAT('SYSTEM ABORTED'/'WE WILL REPEAT')
1046      FORMAT('CALCULATE')
1052      FORMAT('USE LIGHT PEN TO SELECT ITEMS TO BE CORRECTED'/'WAITING FO
         *R YOU TO MAKE A SELECTION'//)
1053      FORMAT('USE LIGHT PEN'/'SELECT PROB. TO BE CORRECTED'/'SELECT CAL
         *CULATE OR STORE WHEN YOU ARE SATISFIED'/'WAITING FOR YOUR SELECTIO
         *N')
42        RETURN
          END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR ICORR
 COMMON  1696  VARIABLES  14  PROGRAM  364

RELATIVE ENTRY POINT ADDRESS IS 00A6 (HEX)

END OF COMPILATION

// DUP

*DELETE             ICORR
CART ID 0021    DB ADDR 2C54     DB CNT 0017

*STORE      WS UA  ICORR
```

```
PAGE 4    ICORR

CART ID 0021     DB ADDR 34F8     DB CNT 0017


PAGE 1    ICRVP

// JOB                                                        ICRVP

LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
  0000        0021        0021         0000

V2 M08    ACTUAL 32K    CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**CORRECT TERM VALUES AND PROB—R DE LANO 3/27/71 REV 05/10/71
```

PAGE 2 ICRVP CORRECT TERM VALUES AND PROB—R DE LANO 3/27/71 REV 05/10/71

```
      SUBROUTINE ICRVP(ICODE,ISECT,IMFLG,YTOP,IVRPB)
C CALLED BY BOTH INITV AND ICORR AFTER SELECTION OF TERMINAL ITEM
C INCLUDES DELETION OF TERM ITEMS AND REVISION OF 'C' ACT FLAGS
      COMMON D(512),C(4,64), B(4,16),A(4,4)
      IDSPL = 1
      IKEYB = 6
      IPLT = 7
      IPRNT = 5
      YINC = .025
      IDS = (ICODE—4)/4
      CALL GIODE(72)
290   PSUM = 0.
      DO 300 I = 4,7
      IDE = I+IDS*4
      CALL GIODE(IDE)
      CALL GIODE(IDE)
300   CONTINUE
      DO 302 IDB = 1,4
      IDD = IDB+4*IDS—4
      ID = IDD+ISECT*64—64
      WRITE(IDSPL,1006)ID
      IF(IMFLG)301,301,3011
301   MARK = 1
3011  IPFLG = 0
      CALL    INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
      YV = YTOP—IDD*.04—YINC
      IBKT = IDB+IDS*4+3
      IF(IMFLG)313,313,311
311   IF(MARK)313,313,312
312   MARK = 0
      YVVV = YV+.02
      CALL FPLOT(—2,9.9,YVVV)
      CALL POINT(1)
313   CALL GIOLB(IBKT)
      CALL FCHAR(8.5,YV,.11,.11,0.)
      WRITE(IPLT,1007)D(ID)
      IPD = 256+ID
```

```
PAGE 3    ICRVP    CORRECT TERM VALUES AND PROB—R DE LANO 3/27/71 REV 05/10/71
      314     WRITE(IDSPL,1021)ID
              READ(IKEYB,1009)D(IPD)
              IF(D(IPD))3140,3141,3141
      3140    WRITE(IDSPL,1048)
              PAUSE 14
              GO TO 314
      3141    CALL GIOLB(IBKT)
              CALL FCHAR(8.,YV,.11,.11,0.)
              WRITE(IPLT,1009)D(IPD)
              CALL GIORB
              PSUM = PSUM+D(IPD)
C PRINT BRANCH NO. AND PROB WHEN VARYING PROB
              IF(IVRPB)302,302,3142
      3142    WRITE(IPRNT,1093)ID,D(IPD)
      302     CONTINUE
              K = ISECT*16—16+IDS
              IF(PSUM)307,3020,303
      3020    C(1,K) = 0.
              GO TO 309
      303     IF(PSUM—.9)307,307,305
      305     IF(PSUM—1.1)3050,307,307
      3050    C(1,K) = 2.
              GO TO 309
      307     WRITE(IDSPL,1010)PSUM
              PAUSE 7
              GO TO 290
      1006    FORMAT('TYPE TERMINAL VALUE FOR BRANCH ',I3,' IF VALUE WOULD EXCEE
             *D 3 DIGITS'/'USE K AFTER DIGITS FOR TIMES 1,000, M FOR TIMES 1,000
             *,000'/'THEN PUSH ALTN CODING AND END AT THE SAME TIME')
      1007    FORMAT(F10.2)
      1009    FORMAT(F5.3)
      1010    FORMAT('THE SUM OF YOUR PROB FOR THIS NODE IS ',F5.3,/'SINCE IT IS
             * NOT BETWEEN .9 AND 1.1 WE WILL REPEAT'/'PUSH PROG START (F/K 31)'
             *)
      1021    FORMAT('TYPE THE PROBABLITY FOR BRANCH',I3,' (IN DECIMAL) WITHOUT
             *SIGN'/'AND PUSH ALTN CODING AND END AT THE SAME TIME'/)
```

```
PAGE 4     ICRVP     CORRECT TERM VALUES AND PROB—R DE LANO 3/27/71 REV 05/10/71

1048       FORMAT('YOU HAVE GIVEN ME A NEGATIVE PROB'/'I CAN ONLY USE POSITIV
          *E VALUES WE WILL REPEAT PUSH START')
1093       FORMAT('THE PROB OF BRANCH— ',I2,' HAS BEEN CHANGED TO 'F5.3)
309        RETURN
           END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR ICRVP
 COMMON  1696  VARIABLES  30  PROGRAM  698

RELATIVE ENTRY POINT ADDRESS IS 0167 (HEX)

END OF COMPILATION

// DUP

*DELETE              ICRVP
CART ID 0021    DB ADDR 2C54    DB CNT 002C

*STORE      WS  UA  ICRVP
CART ID 0021    DB ADDR 34 DF   DB CNT 002C


PAGE 1     INITB

// JOB                                                   INITB

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021         0021        0000

V2 M08     ACTUAL 32K     CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**INITIATE SECOND STAGE PROB—R DE LANO 3/14/71 REV 3/27/71
```

```
PAGE 2      INITB      INITIATE SECOND STAGE PROB—R DE LANO 3/14/71 REV 3/27/71
            SUBROUTINE INITB(XSCLE,YSCLE,XPOST,YPOST,ISTG2,ISTG4,YTOP,BIT,YPOS
           *,YDLT,ICRCT,IVRPB)
C CALLED ONCE BY MAIN
C CONTAINS SECOND STAGE CORRECTION ROUTINES
            COMMON D(512),C(4,64), B(4,16)A(4,4)
            IPLT = 7
            IDSPL = 1
            IKEYB = 6
            ISTRT = 1
            IFINI = 4
            WRITE(IDSPL,1041)
            PAUSE 12
            ISECT = 6
            CALL     ISTGB(ISTRT,IFINI,YTOP,BIT,ISECT,ICODE,IVRPB)
            ISTG2 = 1
            ISTG4 = 1
C POSITION OFF SCREEN TO ERASE
            CALL FPLOT(1,1100.,0.)
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
            CALL     TREE(ISTG2,ISTG4,YTOP,BIT,ICRCT)
109         WRITE(IDSPL,1047)
C WRITE MENU CONTAINING FINI
            CALL GIOLB(72)
            CALL FPLOT(1,0.,YPOS)
            WRITE (IPLT,1048)
            CALL GIORB
C
C LOOP ON GIOLP UNTIL SELECTION
C
1090        CALL GIOLP(ICODE,IX,IY)
            IF(ICODE)1090,1090,1091
1091        IF(ICODE—8)110,111,111
110         WRITE(IDSPL,1024)
            PAUSE 9
            GO TO 109
111         IF(ICODE—72)110,120,112
```

```
PAGE 3     INITB     INITIATE SECOND STAGE PROB—R DE LANO 3/14/71 REV 3/27/71

112        IF(ICODE—88)113,113,110
113        ISTRT = (ICODE—69)/4
           IFINI = ISTRT
C DELETE PROB FOR SELECTED NODE
           DO 114 I = 69,72
           CALL GIODE(I+ISTRT*4)
114        CONTINUE
           CALL     ISTGB(ISTRT,IFINI,YTOP,BIT,ISECT,ICODE,IVRPB)
           GO TO 109
C HERE ONLY WHEN FINI SELECTED
120        CALL GIODE(72)
           RETURN
1024       FORMAT('SYSTEM ABORTED'/'MAKE ANOTHER SELECTION'/
          *'PUSH PROG START (F/K 31) NOW')
1041       FORMAT('WE WILL NOW ASSIGN PROBABLITIES TO THE 16 SECOND STAGE BRA
          *NCHES'/'THESE BRANCHES ARE CALLED B. YOU MAY CORRECT THESE PROB AT
          * THE END'/'ASSIGN 0 PROB TO ELIMINATE THAT AND FOLLOWING TERMINAL
          *BRANCHES'/'TO CONTINUE PUSH PROB START (F/K 31)')
1047       FORMAT('YOU MAY NOW CORRECT STAGE 2 PROB'/'SELECT VALUES TO BE COR
          *RECTED WITH THE LIGHT PEN'/'WHEN FINISHED CORRECTING SELECT FINI'/
          *'WAITING FOR YOUR SELECTION')
1048       FORMAT('FINI')
           END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR INITB
 COMMON  1696  VARIABLES  12  PROGRAM  466

RELATIVE ENTRY POINT ADDRESS IS 010F (HEX)

END OF COMPILATION

// DUP

*DELETE     INITB
```

```
PAGE 4     INITB

CART ID 0021    DB ADDR 2C54    DB CNT 001C

*STORE        WS  UA  INITB
CART ID 0021    DB ADDR 34EB    DB CNT 001C


PAGE 1     INITV

// JOB                                                  INITV

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021         0021         0000

V2 M08    ACTUAL 32K    CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**SECTION TERMINAL VALUES AND PROB — R DE LANO 2/28/71 REV 08/01/71
```

```
PAGE 2     INITV     SECTION TERMINAL VALUES AND PROB — R DE LANO 2/28/71
REV 08/01/71

          SUBROUTINE INITV(ISECT,YTOP,ICODE,IIFLG,YPOS,YDLT,IMFLG,IFLG4,IVRP
         *B)
C CALLED ONCE BY MAIN FOR EACH SECTION.
C OBTAINS TERMINAL VALUES AND THEIR PROB AND DISPLAYS THEM ON TREE
C ASSIGNS ACT FLAGS IN ARRAY 'C'
C CONTAINS CORRECTION ROUTINES DURING INPUT PHASE
C ICODE FROM GIOLP. ICODE LESS THAN 8—RETURN,8 THRU 71—ERASE NODE VALUE AND PROB
C GREATER THAN 71—ERROR MESSAGE
C IMFLG EQUALS 1 FOR*1/1000 SCALE CHANGE. SET IN INPUT AND MAIN2.
C IIFLG DETERMINES NODE DO LOOP START, I2FLG BRANCH DO LOOP START. BOTH SET HERE
C IFLG4 SKIPS INSTRUCTIONS AFTER FIRST TIME. SET HERE.
C YPOS AND YDLT DETERMINE MENU LOCATION AND SPACING
C F/K 8 SKIPS NODES.
          COMMON D(512),C(4,64), B(4,16),A(4,4)
          CALL DATSW(0,ISW)
          CALL GATSW(—1,ISW)
          ISAV3 = 0
          I2FLG = 0
          ISTRT = 1
          IPLT = 7
          IDSPL = 1
          IKEYB = 6
          YINC = .025
          CALL     MENU(YPOS,YDLT)
C NOW TO ADD TERMINAL VALUES AND THEIR PROBABILITIES
C ARRAY D TO CONTAIN (1 TO 256) TERMINAL VALUES AND (257 TO 512) PROB
C SKIP INSTRUCTIONS AFTER FIRST TIME
          IF(IFLG4)4,4,6
4         WRITE(IDSPL,1003)
          WRITE(IDSPL,1004)
          PAUSE 1
          WRITE(IDSPL,1025)
          WRITE(IDSPL,1004)
          PAUSE 2
          WRITE(IDSPL,1008)
          IFLG4 = 1
          PAUSE 3
```

```
PAGE 3    INITV    SECTION TERMINAL VALUES AND PROB — R DE LANO 2/28/71
REV 08/01/71

C SCAN 'C' NODES
6         DO 103 IDS = IIFLG,16
          K = ISECT*16+IDS—16
C I2FLG IS SET (STMT 731+2) BY SELECTING ANOTHER NODE TO BE CORRECTED
          IF(I2FLG)600,600,601
600       PSUM = 0.
C SCAN 'D' BRANCHES
601       DO 101 IDB = ISTRT,4
          CALL GLITE(8,—1)
          IDD = IDB+4*IDS—4
          ID = IDD+ISECT*64—64
C I2FLG IS SET (STMT 731+2) BY SELECTING ANOTHER NODE TO BE CORRECTED
          IF(I2FLG)602,602,8
602       WRITE(IDSPL,1006)ID
          IF(IMFLG)61,61,62
61        MARK = 1
62        IPFLG = 0
          CALL     INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
          YV = YTOP—IDD*.04—YINC
C GIOLB ASSIGNS A NUMBER (8 TO 71)TO EACH GROUP OF TERMINAL NUMBERS
          IBKT = IDB+IDS*4+3
          CALL GIOLB(IBKT)
          CALL FCHAR(8.5,YV,.11,.11,0.)
          WRITE(IPLT,1007)D(ID)
C ERASE SKIP MARKER
          DO 621 J = 1,ISAV3
          CALL GIODE (89)
621       CONTINUE
C HERE TO MARK VALUES AFTER SCALE CHANGE
          IF(IMFLG)65,65,63
63        IF(MARK)65,65,64
64        MARK = 0
          YVVV = YV+.02
          CALL GIORB
          CALL FPLOT(—2,9.9,YVVV)
          CALL POINT (1)
65        IPD = 256+ID
```

```
PAGE 4     INITV     SECTION TERMINAL VALUES AND PROB — R DE LANO 2/28/71
REV 08/01/71

            IIFLG = IDS
            ISTCP = 16—IDS
            DO 67 IADV = 1,ISTOP
            CALL GATSW(—8,IVAL)
            IF(IVAL—2)675,66,675
C HERE TO SKIP NODES WHEN F/K 8 IS DEPRESSED
C
C
C CHECK FOR PROB SUM EITHER 0 OR BETWEEN .9 AND 1.1
C ASSIGN ACT FLAGS
66          IF(PSUM)14,661,662
661         C(1,K) = 0.
            GO TO 665
662         IF(PSUM—.9)14,14,663
663         IF(PSUM—1.1)664,14,14
664         C(1,K) = 2.
            PSUM = 0.
665         IIFLG = IIFLG+1
            K = ISECT*16+IIFLG—16
            YMARK = YTOP—(4*IIFLG—3)*.04
            CALL GIOLB (89)
            CALL FPLOT(—2,7.9,YMARK)
            CALL POINT (5)
            CALL GLITE(8,—1)
            WRITE(IDSPL,1077)
            PAUSE 4
            CALL GATSW(—8,IVAL)
            IF(IVAL—2)675,67,675
67          CONTINUE
675         ISAV3 = IIFLG—IDS
            IF(ISAV3)6,68,6
C
C HAS LIGHT PEN SELECTION OCCURRED
C
68          CALL GIOLP(ICODE, IX,IY)
            IF(ICODE)8,8,7
7           IF(ICODE—8)100,71,71
```

```
PAGE 5     INITV     SECTION TERMINAL VALUES AND PROB — R DE LANO 2/28/71
REV 08/01/71
71          IF(ICODE—72)73,76,74
C
C HERE TO REDO NODE IN RESPONSE TO MENU SELECTION W/O MENU AVAILABLE
C
73          IIFLG = IDS
            ISAVE = ICODE/4
            IF(ISAVE—IDS—1)731,730,730
C HERE WHEN CORRECTING NODE BEING WORKED ON
730         IIFLG = IIFLG+1
            ISTRT = 1
            GO TO 732
C HERE WHEN CORRECTING EARLIER NODE
731         ISTRT = IDB
            I2FLG = 1
732         CALL     ICRVP(ICODE,ISECT,IMFLG,YTOP,IVRPB)
            GO TO 6
74          IF(ICODE—119)76,200,76
C HERE FOR SYSTEM ABORT
76          WRITE(IDSPL,1023)
            GO TO 65
C BACK TO NORMAL NODE INITIATION — READ PROB
8           WRITE(IDSPL,1021)ID
            I2FLG = 0
            READ(IKEYB,1009)D(IPD)
            IF (D(IPD))90,9,9
90          WRITE(IDSPL,1048)
            PAUSE 13
            GO TO 8
9           CALL GIOLB(IBKT)
            CALL FCHAR(8.,YV,.11,.11,0.)
            WRITE(IPLT,1009)D(IPD)
            CALL GIORB
            PSUM = PSUM+D(IPD)
101         CONTINUE
C CHECK FOR PROB SUM EITHER 0 OR BETWEEN .9 AND 1.1
C ASSIGN ACT FLAGS
100         IF(PSUM)14,105,10
```

```
PAGE 6     INITV     SECTION TERMINAL VALUES AND PROB — R DE LANO 2/28/71
REV 08/01/71

105     C(1,K) = 0.
        GO TO 110
10      IF(PSUM—.9)14,14,12
12      IF(PSUM—1.1)106,14,14
106     C(1,K) = 2.
        GO TO 110
14      WRITE(IDSPL,1010)PSUM
C HERE TO ERASE VALUES AND PROB ASSOCIATED WITH NODE BEING WORKED ON
        DO 102 I = 4,7
        CALL GIODE(I+IDS*4)
        CALL GIODE(I+IDS*4)
102     CONTINUE
        DO 104 I = 1,4
        INODE = ID/4
        IERSE = I+INODE*4
        D(IERSE+256) = 0.
        D(IERSE) = 0.
104     CONTINUE
        PAUSE 6
        GO TO 6
C
C HERE WHEN SELECTION WAS A SECTION, STAGE 2 OR ALL SECTIONS
C
110     IF(ICODE)103,200,200
103     CONTINUE
1003    FORMAT('WE WILL START ASSIGNING TERMINAL VALUES AND THEIR PROBABLI-
       *TIES'/'BEGIN WITH THE TOP BRANCH. ASSIGNING 0 PROB WILL ELIMINATE
       *THAT BRANCH'/'NODES HAVING ALL 0 BRANCHES WILL BE ELIMINATED')
1004    FORMAT('TO CONTINUE PUSH PROG START(F/K 31)')
1006    FORMAT('TYPE TERMINAL VALUE FOR BRANCH ',I3,' IF VALUE WOULD EXCEE
       *D 3 DIGITS'/'USE KAFTER DIGITS FOR TIMES 1,000, M FOR TIMES 1,000
       *,000'/'THEN PUSH ALTN CODING AND END AT THE SAME TIME')
1007    FORMAT(F10.2)
1008    FORMAT('PUSH F/K 8 JUST BEFORE GIVING A TERMINAL VALUE TO SKIP NOD
       *ES'/'I WILL SKIP TO THE NEXT NODE AFTER YOU ENTER THE VALUE'/
       *'ZEROS WILL BE ENTERED FOR THE REST OF THE NODE AND THOSE SKIPPED'
```

```
PAGE 7     INITV     SECTION TERMINAL VALUES AND PROB — R DE LANO 2/28/71
REV 08/01/71

           */'SELECT NEW SECTION WHEN NO MORE ACTIVE BRANCHES. PUSH PROG START
           * NOW')
     1009   FORMAT(F5.3)
     1010   FORMAT('THE SUM OF YOUR PROB FOR THIS NODE IS ',F5.3,/'SINCE IT IS
           * NOT BETWEEN .9 AND 1.1 WE WILL REPEAT'/'PUSH PROG START (F/K 31)'
           */)
     1021   FORMAT('TYPE THE PROBABLITY FOR BRANCH',I3,' (IN DECIMAL) WITHOUT
           *SIGN'/'AND PUSH ALTN CODING AND END AT THE SAME TIME'/)
     1023   FORMAT('SYSTEM ABORTED'/'WE WILL REPEAT')
     1025   FORMAT('IF THE VALUE OR PROB DISPLAYED ON THE TREE IS WRONG'/
           *'SELECT THAT VALUE WITH LIGHT PEN'/'I WILL GO BACK TO IT AFTER YOU
           * GIVE ME THE FOLLOWING TERMINAL VALUE')
     1048   FORMAT('YOU HAVE GIVEN ME A NEGATIVE PROB'/'I CAN ONLY USE POSITIV
           *E VALUES WE WILL REPEAT PUSH START')
     1077   FORMAT('PUSH F/K 8 AND F/K 31 TO SKIP ANOTHER NODE'/'JUST F/K 31 T
           *O CONTINUE'//
     200    RETURN
            END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR INITV
 COMMON  1696  VARIABLES  44  PROGRAM  1488

RELATIVE ENTRY POINT ADDRESS IS 02E2 (HEX)

END OF COMPILATION

// DUP

*DELETE              INITV
CART ID 0021     DB ADDR 2C54     DB CNT 005B

*STORE        WS  UA  INITV
CART ID 0021     DB ADDR 34B7     DB CNT 005B
```

```
PAGE 1      INPUT

// JOB                                                                    INPUT

  LOG DRIVE    CART SPEC    CART AVAIL    PHY DRIVE
    0000         0021         0021          0000

V2 M08      ACTUAL 32K    CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**READ DOLLARS AND ADJUST MATRIX D FOR M—R DE LANO 3/09/71 REV 3/10/71
```

```
PAGE 2      INPUT      READ DOLLARS AND ADJUST MATRIX D FOR M—R DE LANO 3/09/71
REV 3/10/71

            SUBROUTINE INPUT(DLRS,MAXD,MIND,ID,IMFLG,IPFLG)
C CALLED BY INITV, ASPRF,ICRVP
C READS VALUES FROM KEYBOARD FOR INITV AND ASPRF
C CHECKS FOR LEGAL CHARACTERS,     MORE THAN 6 CHARACTERS AND
C CHARACTERS AFTER K OR M
C CHECKS FOR MORE THAN THREE DIGITS
C COMPACTS AROUND IMBEDDED BLANKS
C SCALES VALUES IN ARRAY D AND FOLLOWING VALUES BY 1/1000 WHEN M IS USED
C DETERMINES MAX AND MIN
C STORES VALUE IN ARRAY D
C IMFLG INDICATES M USED IN INPUT. SET HERE AND IN MAIN. STORED IN B(1,2)
C WHEN PROB STORED ON DISK
C IPFLG INDICATES THIS SUBROUTINE CALLED BY ASPRF
C IFLG3 COUNTS DECIMAL POINT AND NUMBER OF DIGITS THAT FOLLOW
            INTEGER V(7),SIGN,H(17)
            REAL MAXD,MIND
            COMMON D(512),C(4,64), B(4,16),A(4,4)
            DATA H/'0','1','2','3','4','5','6','7','8','9',' ','.','+','—','K'
           *,'M','$'/
            SIGN = 1
            IDSPL = 1
            IKEY = 6
            CALL GIOEP
            CALL DATSW(0,ISW)
            ICTR = 0
            IFLG3 = 0
            DLRS = 0.
            READ(IKEY,1014)V
            DO 30 I = 1,6
            DO 5 J = 1,17
            K = I
C CHECK FOR NONLEGAL CHARACTER
            IF(V(I)—H(J))5,10,5
     5      CONTINUE
            WRITE(IDSPL,1040)V(I)
            GO TO 3
C IS CHARACTER NUMERIC OR SPECIAL
```

```
PAGE 3     INPUT      READ DOLLARS AND ADJUST MATRIX D FOR M—R DE LANO 3/09/71
REV 3/10/71

C CLOSE IMBEDDED BLANKS
10        IF(J—11)14,30,15
14        ICTR = ICTR+1
C CHECK FOR MORE THAN THREE DIGITS
          IF(ICTR—3)141,141,52
141       IF(IFLG3)142,142,143
142       DLRS = DLRS*10+J—1
          GO TO 30
143       DLRS = DLRS+(J—1)/10.**IFLG3
          IFLG3 = 1+IFLG3
          GO TO 30
C HERE FOR SPECIAL
15        IF(V(I)—H(12))16,25,16
16        IF(V(I)—H(13))17,30,17
17        IF(V(I)—H(14))18,21,18
18        IF(V(I)—H(15))19,22,19
19        IF(V(I)—H(16))20,23,20
20        IF(V(I)—H(17))34,30,34
21        SIGN = —1
          GO TO 30
C HERE IF LAST CHARACTER THAT WILL BE RECOGNIZED WAS K
22        DLRS = DLRS*1000.
          GO TO 47
C HERE IF LAST CHARACTER THAT WIL BE RECOGNIZED WAS M
23        DLRS = DLRS*1000
          WRITE(IDSPL,1044)
          PAUSE 10
          IF(ID—1)34,40,42
25        IFLG3 = 1
30        CONTINUE
C THRU DO LOOP WITH LAST BLANK AND NO K OR M
          GO TO 47
C HERE FOR SYSTEM ERROR
34        WRITE(IDSPL,1043)V(I)
          PAUSE 8
          GO TO 3
C HERE FOR M AND FIRST VALUE IN ARRAY
```

```
PAGE 4      INPUT      READ DOLLARS AND ADJUST MATRIX D FOR M—R DE LANO
3/09/71  REV 3/10/71

40        IMFLG = 1
          GO TO 50
C HERE IF M, DIVIDE PREVIOUS VALUES IN ARRAY D BY 1000
42        IF(IMFLG)34,43,50
43        ISTOP = ID—1
          DO 44 I = 1,ISTOP
          D(I) = D(I)/1000.
44        CONTINUE
          IMFLG = 1
          MAXD = MAXD/1000.
          MIND = MIND/1000.
          GO TO 50
C CHECK FOR PREVIOUS M
47        IF(IMFLG)34,50,48
48        DLRS = DLRS/1000.
50        DLRS = SIGN*DLRS
          IF(ID—1)34,49,51
49        MAXD = DLRS
          MIND = DLRS
C HERE TO CHECK FOR EXTRA DIGITS
51        IF(V(K+1)—H(11))52,54,52
52        IF(IPFLG)53,53,530
53        WRITE(IDSPL,1020)ID
          GO TO 3
530       WRITE(IDSPL,1080)
          GO TO 3
54        IF(IPFLG)540,540,60
540       D(ID) = DLRS
C HERE TO FIND MAX AND MIN
          IF(MAXD—DLRS)55,60,56
55        MAXD = DLRS
          GO TO 60
56        IF(DLRS—MIND)58,60,60
58        MIND = DLRS
60        RETURN
1014      FORMAT(7A1)
1020      FORMAT('I ASKED FOR NOT MORE THAN THREE DIGITS'/'LETS START AGAIN'
```

```
PAGE 5      INPUT      READ DOLLARS AND ADJUST MATRIX D FOR M—R DE LANO 3/09/71
REV 3/10/71

            */'WRITE THE VALUE FOR BRANCH NO',13' USING ONLY THREE DIGITS')
1040          FORMAT ('I DO NOT RECOGNIZE CHARACTER 'A1,' USE ONLY 0 THRU 9,+,−
            *,K, OR M'/'NOT MORE THAN THREE DIGITS, PLEASE'/'AND NO IMBEDDED BL
            *ANKS, WE WILL START AGAIN')
1043        FORMAT('SYSTEM ERROR, ASK FOR HELP V(I) IS ',A1)
1044          FORMAT('YOUR VALUES ARE BEING DIVIDED BY 1000'/'PLEASE CONTINUE AS
            * BEFORE'/'PUSH PROG START (F/K 31) TO CONTINUE')
1080          FORMAT('I ASKED FOR NOT MORE THAN THREE DIGITS'/'LETS START AGAIN'
            */)
              END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR INPUT
 COMMON  1696  VARIABLES  40  PROGRAM  800

RELATIVE ENTRY POINT ADDRESS IS 0134 (HEX)

END OF COMPILATION

// DUP

*DELETE                  INPUT
CART ID 0021       DB ADDR 2C54       DB CNT 0037

*STORE          WS  UA  INPUT
CART ID 0021       DB ADDR 34D2       DB CNT 0037
```

```
PAGE 1      ISTGB

// JOB                                                              ISTGB

  LOG DRIVE    CART SPEC    CART AVAIL    PHY DRIVE
    0000         0021          0021          0000

V2 M08   ACTUAL 32K   CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**READ STAGE 2 PROB—R DE LANO 3/27/71 REV 06/09/71
```

PAGE 2 ISTGB READ STAGE 2 PROB—R DE LANO 3/27/71 REV 06/09/71

```
            SUBROUTINE ISTGB(ISTRT,IFINI,YTOP,BIT,ISECT,ICODE,IVRPB)
C CALLED BY ICORR AND TWICE BY INITB
C ISTG2 AND ISTG4 ARE 1 WHEN TREE IS TO BE PRUNED
C INSERTS AND CORRECTS 'A' AND 'B' ACT EVENT FLAGS
      COMMON D(512),C(4,64), B(4,16),A(4,4)
      IKEYB =6
      IPLT = 7
      IDSPL = 1
      IPRNTN = 5
      HLFY = YTOP/2
      DO 400 IPB = ISTRT,IFINI
320   PSUM = 0.
      DO 390 IBPB = 1,4
      IB = IBPB+4*IPB—4
321   WRITE(IDSPL,1046)IB
      SAVE = B(2,IB)
      READ(IKEYB,1009)B(2,IB)
      IF(B(2,IB))322,323,323
322   WRITE(IDSPL,1048)
      GO TO 321
323   IF(ISECT—4)325,325,330
C HERE TO CORRECT WHEN TREE IS MAGNIFIED
325   IPBB = 1
      GO TO 335
330   IPBB = IPB
335   YV = YTOP—ABPB*.37—IPBB*2.56+2.07+HLFY/BIT
      IBKT = IB+72
C GIOLB ASSIGNS A NUMBER (73 TO 88) TO THE PROB WRITTEN ON EACH BRANCH
      CALL GIOLB(IBKT)
      CALL FCHAR(3.0,YV,.1,.1,0.)
      WRITE(IPLT,1009)B(2,IB)
      CALL GIORB
      PSUM = PSUM+B(2,IB)
C PRINT BRANCH NO. AND PROB WHEN VARYING PROB
      IF(IVRPB)390,390,350
350   WRITE(IPRNT,1092)IB,SAVE,B(2,IB)
```

```
PAGE 3   ISTGB   READ STAGE 2 PROB—R DE LANO 3/27/71 REV 06/09/71

390     CONTINUE
        IF(PSUM—0)393,3900,391
C DELETE ACT/EVENT FLAG
3900    B(1,4*IPB—3)=0.
        A(1,IPB)=0.
        GO TO 400
391     IF(PSUM—.9)393,393,392
C SET ACT/EVENT FLAG
392     IF(PSUM—1.1)3920,393,393
3920    B(1,4*IPB—3)=1.
        A(1,IPB)=2.
        GO TO 400
393     WRITE(IDSPL,1010)PSUM
        DO 395 I=69,72
        CALL GIODE(I+IPB*4)
395     CONTINUE
        PAUSE 8
        GO TO 320
400     CONTINUE
1009    FORMAT(F5.3)
1010    FORMAT('THE SUM OF YOUR PROB FOR THIS NODE IS ',F5.3,/'SINCE IT IS
       * NOT BETWEEN .9 AND 1.1 WE WILL REPEAT'/'PUSH PROG START (F/K 31)'
       *)
1046    FORMAT('TYPE THE PROB FOR BRANCH B—',12,/'AND PUSH BOTH ALTN CODIN
       *G AND END'/)
1048    FORMAT('YOU HAVE GIVEN ME A NEGATIVE PROB'/'I CAN ONLY USE POSITIV
       *E VALUES WE WILL REPEAT PUSH START')
1092    FORMAT('THE PROB OF BRANCH B— ',12,' HAS BEEN CHANGED FROM ',
       *F5.3,' TO ',F5.3)
        RETURN
        END

FEATURES SUPPORTED
  ONE WORD INTEGERS

CORE REQUIREMENTS FOR ISTGB
 COMMON  1696  VARIABLES  24  PROGRAM  536
```

```
PAGE 4    ISTGB    READ STAGE 2 PROB—R DE LANO 3/27/71 REV 06/09/71

RELATIVE ENTRY POINT ADDRESS IS 00EF (HEX)

END OF COMPILATION

// DUP

*DELETE                 ISTGB
CART ID 0021    DB ADDR 2C54    DB CNT 0023

*STORE        WS  UA  ISTGB
CART ID 0021    DB ADDR 34E9    DB CNT 0023



PAGE 1    MAIN
                                                    MAIN
// JOB

  LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
    0000        0021        0021         0000

V2 M08  ACTUAL 32K  CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**MAIN DATA PROGRAM — R DE LANO 2/21/71 REV 08/01/71
```

```
PAGE 2    MAIN    MAIN DATA PROGRAM— R DE LANO 2/21/71 REV 08/01/71
      SUBROUTINE MAIN(MAXD,MIND,IRSME,ISTG2,ISTOR,ICRCT,IFLG4,IVRPB,
     *IMFLG)
C CALLED ONCE BY MAIN2 FOR NEW PROBLEM, TO RESUME OLD, TO CORRECT
C OBTAINS INPUT DATA FROM DM AND PROVIDES CORRECTION FACILITIES
C ICRCT IS SET HERE AND IN MAIN2 TO PROVIDE CORRECTION DISPLAY
      REAL MAXD,MIND
      COMMON D(512),C(4,64), B(4,16),A(4,4)
      IKEYB = 6
      IPLT = 7
      IDSPL = 1
      YTOP = 11.
      BIT = 55.
      YAAA = 9.695
      XSCLE = 1.
      YSCLE = .9
      YDLT = .36/YSCLE
      XPOST = 0.
      YPOST = 0.
      YPOS = 9.72/YSCLE+YPOST
C SKIP TO DISPLAY OF MENU AND TREE WHEN RESUMING PROBLEM OR CORRECTING
      IF(ICRCT)3,3,9
3     IF(IRSME)5,5,92
5     IMFLG = 0
C ZERO ARRAYS FOR NEW PROBLEM
      DO 51 M = 1,512
      D(M) = 0
51    CONTINUE
      DO 52 N = 1,64
      DO 52 M = 1,4
      C(M,N) = 0
52    CONTINUE
      DO 54 N = 1,4
      DO 54 M = 1,16
      B(N,M) = 0
54    CONTINUE
      DO 56 N = 1,4
      DO 56 M = 1,4
```

```
PAGE 3    MAIN    MAIN DATA PROGRAM— R DE LANO 2/21/71 REV 08/01/71

            A(M,N)=0
56          CONTINUE
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
            ISTG2=0
            ISTG4=0
            ICRCT=0
            CALL        TREE(ISTG2,ISTG4,YTOP,BIT,ICRCT)
            WRITE(IDSPL,1005)
            WRITE(IDSPL,1011)
            WRITE(IDSPL,1012)
C HERE TO ENABLE MENU SELECTION — MAGNIFIED SECTION, ALL SECTION, STAGE 2,
C STORE
9           YPOS=9.72/YSCLE+YPOST
            YDLT=.36/YSCLE
            CALL        MENU(YPCS,YDLT)
            CALL GIORB
C DELETE 'STAGE 2' FROM MENU WHEN CORRECTING
            IF(ICRCT)7,7,6
6           CALL GIODE(7)
7           WRITE(IDSPL,1019)
C
C LOOP ON GIOLP UNTIL SELECTION
C
10          CALL GIOLP(ICODE,IX,IY)
C HERE WHEN SELECTION MADE
C    ISECT   TELLS   WHICH   SECTION   OF   TREE   IS   TO   BE   DISPLAYED
11          ISECT=ICODE
            IF(ISECT)10,10,20
20          IF(ISECT—5)30,910,90
C MAGNIFICATION ROUTINES
C POSITION BEAM OFF SCREEN TO ERASE
30          CALL FPLOT(1,1100.,0.)
            YSCLE=3.9
            YPOST=8.45
            YPOS=9.72/YSCLE+YPOST
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
```

```
PAGE 4    MAIN    MAIN DATA PROGRAM— R DE LANO 2/21/71 REV 08/01/71

C WRITE NEW FIRST NODE AND BRANCHES
          YA = 9.67
          CALL FCHAR(0.,YA,.1,.1,0.)
          WRITE(IPLT,1000)
C WRITE SCALE CHANGE MARKER IF PREVIOUS VALUES CONTAINED M
          IF(IMFLG)35,35,32
32        CALL FPLOT(—2,9,9,YPOS+.05)
          CALL POINT(1)
C SELECT LOOP TO WRITE FIRST FOUR BRANCHES
35        GO TO (40,50,60,70),ISECT
40        DO 102 IA = 1,4
          YA = YAAA—IA*.06+.06
          CALL FPLOT(1,.1,YAAA)
          CALL FPLOT(2,1.98,YA)
102       CONTINUE
          GO TO 75
50        DO 104 IA = 1,4
          YA = YAAA—IA*.06+.12
          CALL FPLOT(1,.1,YAAA)
          CALL FPLOT(2,1.98,YA)
104       CONTINUE
          GO TO 75
60        DO 106 IA = 1,4
          YA = YAAA—IA*.06+.18
          CALL FPLOT(1,.1,YAAA)
          CALL FPLOT(2,1.98,YA)
106       CONTINUE
          GO TO 75
70        DO 108 IA = 1,4
          YA = YAAA—IA*.06+.24
          CALL FPLOT(1,.1,YAAA)
          CALL FPLOT(2,1.98,YA)
108       CONTINUE
C HERE TO DISPLAY MAGNIFIED REST OF TREE
75        BIT = 440.
          YAA = 9.695
          IA = 1
```

PAGE 5 MAIN MAIN DATA PROGRAM— R DE LANO 2/21/71 REV 08/01/71

```
          CALL FCHAR(1.95,9.670,.1,.1,0.)
          WRITE(IPLT,1001)
          ICTR=0
          CALL      PTREE(IA,YTOP,BIT,ISTG2,ISTG4,ICRCT,YAA,ISECT,ICTR)
          IIFLG=1
          YDLT=.36/YSCLE
          CALL      MENU(YPOSL,YDLT)
          IF(ICRCT)76,76,751
751       CALL GIODE(7)
          CALL      ICORR(ISECT,YTOP,ICODE,YPOS,YDLT,IMFLG,ICRCT,IVRPB,BIT)
          GO TO 11
76        CALL      INITV(ISECT,YTOP,ICODE,IIFLG,YPOS,YDLT,IMFLG,IFLG4,IVRP
         *B)
C HERE TO DECODE SELECTION AT BEGINING OF MAIN AND IN INITV
          IF(ICODE)9,9,77
77        IF(ICODE—6)11,92,90
90        IF(ICODE—7)92,93,900
900       IF(ICODE—73)910,901,901
901       IF(ICODE—88)902,902,91
C HERE TO CHANGE STAGE 2 PROB
902       BIT=55
          CALL      ICORR(ISECT,YTOP,ICODE,YPOS,YDLT,IMFLG,ICRCT,IVRPB,BIT)
          GO TO 11
91        IF(ICODE—90)910,999,911
911       IF(ICODE—119)910,912,9111
9111      IF(ICODE—120)910,9112,910
C HERE TO CORRECT
9112      ICRCT=1
          RETURN
C HERE TO STORE PROBLEM ON DISK
912       ISTOR=1
          ICRCT=0
          RETURN
C HERE FOR SYSTEM ERROR
910       WRITE(IDSPL,1024)
          GO TO 9
C HERE TO WRITE TREE W/O TERMINAL ZEROS, ICRCT IS SAVED
```

```
PAGE 6      MAIN      MAIN DATA PROGRAM R DE LANO 2/21/71 REV 08/01/71
92          BIT=55
            ISAVE=0
            IF(ICRCT)924,924,922
922         ISAVE=ICRCT
            ICRCT=0
            WRITE(IDSPL,1094)
924         ISTG4=1
            XSCLE=1.
            XPOST=0.
            YSCLE=.9
            YDLT=.36/YSCLE
            YPOST=0.
            YPOS=9.72/YSCLE+YPOST
C POSITION BEAM OFF SCREEN TO ERASE
            CALL FPLOT(1,1100.,0.)
            CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
            CALL        TREE(ISTG2,ISTG4,YTOP,BIT,ICRCT)
            ICRCT=ISAVE
            YPOSN=YPOS—7*YDLT
            CALL GIOLB(90)
            CALL FPLOT(1,0.,YPCSN)
            WRITE(IPLT,1046)
            CALL GIOLB(120)
            YPOSN=YPOS—8*YDLT
            CALL FPLOT(1,0.,YPOSN)
            WRITE(IPLT,1065)
            CALL GIORB
            GO TO 9
C HERE TO INPUT STAGE 2 PROB
93          BIT=55
            ISTG4=1
            YSCLE=.9
            YDLT=.36/YSCLE
            YPOST=0.
            YPOS=9.72/YSCLE+YPOST
C POSITION BEAM OFF SCREEN TO ERASE
```

```
PAGE 7    MAIN    MAIN DATA PROGRAM— R DE LANO 2/21/71 REV 08/01/71

          CALL FPLOT(1,1100.,0.)
          CALL SCALF(XSCLE,YSCLE,XPOST,YPOST)
          CALL      TREE(ISTG2,ISTG4,YTOP,BIT,ICRCT)
          CALL      INITB(XSCLE,YSCLE,XPOST,YPOST,ISTG2,ISTG4,YTOP,BIT,YPOS
         *,YDLT,ICRCT,IVRPB)
          ICRCT=1
          ISTG2=1
          WRITE(IDSPL,1045)
          CALL GIODE (7)
          CALL GIODE (72)
          YPOSN=YPOS—7*YDLT
          CALL GIOLB(90)
          CALL FPLOT(1.0.,YPOSN)
          WRITE(IPLT,1046)
          CALL GIORB
          GO TO 9
C HERE TO CALCULATE—IE. RETURN
999       WRITE(IDSPL,1047)
          ICRCT=0
          CALL GIODE(89)
1000      FORMAT('O')
1001      FORMAT('O')
1005      FORMAT('THIS IS YOUR INITIAL DECISION TREE')
1011      FORMAT('THERE ARE FOUR EQUAL SECTIONS OF THE TREE FROM TOP TO BOTT
         1OM')
1012      FORMAT('USE LIGHT PEN TO SELECT A SECTION TO BE MAGNIFIED')
1019      FORMAT('WAITING FOR YOU TO MAKE A SELECTION')
1024      FORMAT('SYSTEM ABORTED. MAKE ANOTHER SELECTION'/
         *'TO CORRECT TERMINAL PROB OR VALUES SELECT A SECTION'/
         *'SELECT STAGE 2 PROB TO CORRECT IT')
1045      FORMAT('BEFORE I START COMPUTING THE VALUE OF YOUR TREE'/'PLEASE L
         *OOK AT EACH SECTION AND REVIEW ALL THE VALUES AND PROB'/
         *     'CORRECT BY LIGHT PEN SELECTION.'' SELECT CALCULATE
         * WHEN FINISHED')
1046      FORMAT('CALCULATE')
1047      FORMAT('THANK YOU FOR YOUR INPUT'/'I SHALL NOW COMPUTE AND DISPLAY
```

```
PAGE  8     MAIN     MAIN  DATA  PROGRAM—  R  DE  LANO  2/21/71  REV  08/01/71

              * YOUR RESULTS'//)
 1065          FORMAT('CORRECT')
 1094          FORMAT(//'SELECT THE STAGE 2 PROB YOU WISH TO CHANGE')
               RETURN
               END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR MAIN
 COMMON  1696  VARIABLES  44  PROGRAM  1386

RELATIVE ENTRY POINT ADDRESS IS 01C2 (HEX)

END OF COMPILATION

// DUP

*DELETE                MAIN
CART ID 0021      DB ADDR 2C54   DB CNT 0058

*STORE         WS  UA  MAIN
CART ID 0021      DB ADDR 34AC   DB CNT 0058
```

```
PAGE 1     MENU

// JOB                                                      MENU

 LOG DRIVE    CART SPEC    CART AVAIL    PHY DRIVE
   0000         0021          0021          0000

V2 M08    ACTUAL 32K    CONFIG 32K

// FOR
*ONE WORD INTEGERS
**LIST SOURCE PROGRAM
**MENU — R DE LANO 2/28/71 REV 05/08/71
```

```
PAGE 2     MENU     MENU — R DE LANO 2/28/71 REV 05/08/71

          SUBROUTINE MENU(YPOS,YDLT)
C CALLED BY INITV AND TWICE BY MAIN
C MENU DISPLAYS SECTIONS 1 THRU 4, WRITE TREE, STAGE 2, STORE
          IPLT=7
          YSAVE=YPOS
          CALL GIOLB (1)
          CALL FPLOT(1,0.,YPOS)
          WRITE(IPLT,1013)
          CALL GIOLB(2)
          YPOS=YPOS—YDLT
          CALL FPLOT(1,0.,YPOS)
          WRITE(IPLT,1014)
          CALL GIOLB(3)
          YPOS=YPOS—YDLT
          CALL FPLOT(1,0.,YPOS)
          WRITE(IPLT,1015)
          CALL GIOLB(4)
          YPOS=YPOS—YDLT
          CALL FPLOT(1,0.,YPOS)
          WRITE(IPLT,1016)
          CALL GIOLB(6)
          YPOS=YPOS—YDLT
          CALL FPLOT(1,0.,YPOS)
          WRITE(IPLT,1018)
          CALL GIOLB(7)
          YPOS=YPOS—YDLT
          CALL FPLOT(1,0.,YPOS)
          WRITE(IPLT,1019)
          CALL GIOLB(119)
          YPOSN=YPOS—YDLT
          CALL FPLOT(1,0.,YPOSN)
          WRITE(IPLT,1064)
          CALL GIORB
          YPOS=YSAVE
1013      FORMAT('SECTION 1')
1014      FORMAT('SECTION 2')
```

```
PAGE 3      MENU      MENU — R DE LANO 2/28/71 REV 05/08/71

1015         FORMAT('SECTION 3')
1016         FORMAT('SECTION 4')
1018         FORMAT('ALL SECTIONS')
1019         FORMAT('STAGE 2')
1064         FORMAT('STORE')
             RETURN
             END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR MENU
 COMMON   0  VARIABLES   6   PROGRAM   222

RELATIVE ENTRY POINT ADDRESS IS 003E (HEX)

END OF COMPILATION

// DUP

*DELETE                MENU
CART ID 0021     DB ADDR 2C54      DB CNT 0010

*STORE        WS  UA  MENU
CART ID 0021     DB ADDR 34F4      DB CNT 0010


PAGE 1      PICRD

// JOB                                                   PICRD

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021        0021         0000

V2 M08     ACTUAL 32K     CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**DRAW AND LABEL PREF AXES R DE LANO— 5/9/71
```

PAGE 2 PICRD DRAW AND LABEL PREF AXES R DE LANO— 5/9/71

```
      SUBROUTINE PICRD(PI0,PI100)
C CALLED BY ASPRF
      DIMENSION SLBL(7)
      COMMON D(512),C(4,64), B(4,16),A(4,4)
      DATA SLBL/'.0','.2','.4','.6','.8','1.','0.'/
      IPLT = 7
      CALL SCALF(1.,.9,0.,—2.)
      XSTRT = 1.15
      XINC = 1.
C LABEL X AXIS
      XSCLE = (PI100—PI0)/4
      DO 8 I = 1,5
      CALL FCHAR(XSTRT,.65,.1,.1,0.)
      XS = XSCLE*I—XSCLE+PI0
      WRITE(IPLT,1007)XS
      XSTRT = XSTRT+XINC
8     CONTINUE
      CALL FCHAR(6.5,1.0,.1,.1,0.)
      WRITE(IPLT,1086)
      YSTRT = .85
      YINC = 1.
      YSCLE = 1.
C LABEL Y AXIS
      DO 10 I = 1,6
      CALL FCHAR(1.4,YSTRT,.1,.1,0.)
      WRITE(IPLT,1009)SLBL(I)
      YSTRT = YSTRT+YINC
10    CONTINUE
      CALL FGRID(0,2.,.9,1.,4)
      CALL FGRID(1,2.,.9,1.,5)
      CALL FCHAR(2.2,6.10,.1,.1,0.)
      WRITE(IPLT,1085)
C AXES DRAWN AND LABELED
C MARK 0 ON AXIS
      X0 = 2.—PIO*XINC/XSCLE
      CALL FPLOT(—2,X0,.9)
      CALL POINT(0)
```

```
PAGE 3     PICRD     DRAW AND LABEL PREF AXES R DE LANO— 5/9/71

                X0 = X0—.05
                CALL FCHAR(X0,.50,.1,.1,0.)
                WRITE (IPLT,1009)SLBL(7)
      C PLOT A STRAIGHT LINE
                CALL FPLOT(1,2.,.9)
                CALL FPLOT(2,6.,5.9)
      1007      FORMAT(F10.2)
      1009      FORMAT(A4)
      1085      FORMAT('UTILES')
      1086      FORMAT('VALUE')
                RETURN
                END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR PICRD
 COMMON  1696  VARIABLES  38  PROGRAM  282

RELATIVE ENTRY POINT ADDRESS IS 005B (HEX)

END OF COMPILATION

// DUP

*DELETE            PICRD
CART ID 0021    DB ADDR  2C54    DB CNT  0016

*STORE      WS UA PICRD
CART ID 0021    DB ADDR  34F4    DB CNT  0016
```

```
PAGE 1      PINVT
                                                      PINVT
// JOB
  LOG DRIVE    CART SPEC    CART AVAIL    PHY DRIVE
    0000         0021          0021          0000

V2  M08     ACTUAL 32K    CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**INVERT 'A' AND 'C' ARRAYS R DE LANO 5/70 REV. 08/09/71
```

PAGE 2 PINVT INVERT 'A' AND 'C' ARRAYS R DE LANO 5/70 REV. 08/09/71

```
          SUB ROUTINE PINVT(PI0,PI25,PI50,PI75,PI100,PSHFT)
C OPERATES ON COLUMN 4 OF MATRICES
C PINVT TRANSFORMS EXPECTED PI'S IN ARRAYS 'A' AND 'C' BACK TO SAME DIMENSION AS
C ORIGINAL TERMINAL VALUES USING LINEAR INTERPOLATION.
          COMMON D(512),C(4,64), B(4,16),A(4,4)
          IPRNT=5
          IDSPL=1
C TRANSFORM 'A' VALUES— CONDITIONED ON ACT/EVENT FLAG
          DO 16 I=1,4
          K=I
          IF(A(1,I))16,16,2
C IF A(4,I) IS ZERO, IT IS LEGAL IF ALL A(4)S ARE ZERO
2         IF(A(4,I))204,200,204
200       ICTR=0
          DO 202 J=1,4
          IF(A(4,J))16,201,16
201       ICTR=ICTR+1
202       CONTINUE
          IF(ICTR—4)16,203,16
203       A(4,I)=PI0
          GO TO 19
204       IF(A(4,I)—.25)4,4,6
4         A(4,I)=A(4,I)*(PI25—PI0)/.25+PI0
          GO TO 19
6         IF(A(4,I)—.50)8,8,10
8         A(4,I)=A(4,I)*(PI50—PI125)/.25—PI50+2*PI25
          GO TO 19
10        IF(A(4,I)—.75)12,12,14
12        A(4,I)=A(4,I)*(PI75—PI50)/.25—2*PI75+3*PI50
          GO TO 19
14        A(4,I)=A(4,I)*(PI100—PI75)/.25—3*PI100+4*PI75
          GO TO 19
16        CONTINUE
          WRITE(IDSPL,1095)
          WRITE(IPRNT,1095)
          PAUSE 10
19        A(4,K)=A(4,K)+PSHFT
```

```
PAGE 3      PINVT      INVERT 'A' AND 'C' ARRAYS R DE LANO 5/70 REV. 08/09/71

C TRANSFORM 'C' VALUES— CONDITIONED ON ACT/EVENT FLAG
          DO 40 L = 1,64,4
          DO 36 M = 1,4
          I = L+M—1
          IF(C(1,I))36,36,22
C IF C(4,I) IS ZERO, IT IS LEGAL IF ALL C(4)S ARE ZERO
22        IF(C(4,1))224,220,224
220       ICTR = 0
          DO 222 K = 1,4
          IF(C(4,K))36,221,36
221       ICTR = ICTR+1
222       CONTINUE
          IF(ICTR—4)36,223,36
223       C(4,I) = PI0
          GO TO 39
224       IF(C(4,I)—.25)24,24,26
24        C(4,I) = C(4,I)*(PI25—PI0)/.25+PI0
          GO TO 39
26        IF(C(4,I)—.50)28,28,30
28        C(4,I) = C(4,I)*PI50—PI25)/.25—PI50+2*PI25
          GO TO 39
30        IF(C(4,I)—.75)32,32,34
32        C(4,I) = C(4,I)*(PI75—PI50)/.25—2*PI75+3*PI50
          GO TO 39
34        C(4,I) = C(4,I)*(PI100—PI75)/.25—3*PI100+4*PI75
          GO TO 39
36        CONTINUE
          GO TO 40
39        C(4,I) = C(4,I)+PSHFT
40        CONTINUE
1095      FORMAT(' SYSTEM ERROR'/' THERE ARE NO ACT FLAGS IN ARRAY"A"'/
         *' ASK FOR ASSISTANCE'/)
          RETURN
          END

FEATURES SUPPORTED
 ONE WORD INTEGERS
```

```
PAGE 4     PINVT     INVERT 'A' AND 'C' ARRAYS R DE LAND 5/70 REV. 08/09/71

CORE REQUIREMENTS FOR PINVT
 COMMON  1696  VARIABLES  14  PROGRAM  642

RELATIVE ENTRY POINT ADDRESS IS 0047 (HEX)

END OF COMPILATION

// DUP

*DELETE              PINVT
CART ID 0021     DB ADDR 2C54     DB CNT 002E

*STORE       WS  UA  PINVT
CART ID 0021     DB ADDR 34DA     DB CNT 002E
```

```
PAGE 1      PREFR
                                                            PREFR
// JOB
 LOG DRIVE    CART SPEC   CART AVAIL   PHY DRIVE
   0000          0021        0021         0000

V2 M08     ACTUAL 32K     CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
          SUBROUTINE PREFR(PI0,PI25,PI50,PI75,PI100,PSHFT)
C PREFR TRANSFORMS THE TERMINAL VALUES IN ARRAY 'D' INTO PREFERENCE VALUES 'PI'
          COMMON D(512),C(4,64), B(4,16),A(4,4)
          IDSPL = 1
          IPNT = 5
          PSHFT = 0.
          IF(PI0—0.)11,111,111
C HERE TO SHIFT ORIGIN FOR PI WHEN PI0 IS NEGATIVE
11        PSHFT = PI0
          PI0 = 0.
          PI25 = PI25—PSHFT
          PI50 = PI50—PSHFT
          PI75 = PI75—PSHFT
          PI100 = PI100—PSHFT
111       DO 26 N = 1,256
C HERE TO SHIFT ORIGIN FOR TERMINAL VALUES WHEN PI0 IS NEGATIVE
          D(N) = D(N)—PSHFT
          IF(PI0+D(N))2,4,4
2         WRITE(IPNT),1010(D(N),PI0
          WRITE(IDSPL,1010)D(N),PI0
          RETURN
C HERE TO TRANSFORM 'D'.
4         IF(D(N)—PI25)6,6,8
6         D(N) = .25*D(N)/(PI25—PI0)+PI0*.25/(PI25—PI0)
          GO TO 26
8         IF(D(N)—PI50)10,10,12
```

```
PAGE 2     PREFR

10      D(N) = .25*D(N)/(PI50—PI25)+.25—.25*PI25/(PI50—PI25)
        GO TO 26
12      IF(D(N)—PI75)14,14,16
14      D(N) = .25*D(N)/(PI75—PI50)+.50—.25*PI50/(PI75—PI50)
        GO TO 26
16      IF(D(N)—PI100)18,18,20
18      D(N) = .25*D(N)/(PI100—PI75)+.75—.25*PI75/(PI100—PI75)
        GO TO 26
C CHECK FOR TERMINAL VALUE GREATER THAN PI100
20      WRITE(IPNT,1010)D(N),PI100
        WRITE(IDSPL,1010)D(N),PI100
        RETURN
26      CONTINUE
C HERE TO ZERO COLUMN THREE OF ARRAYS C, B AND A.
        DO 110 N = 1,64
        C(3,N) = 0.
110     CONTINUE
        DO 114 M = 1,16
        B(3,M) = 0.
114     CONTINUE
        DO 115 N = 1,4
        A(3,N) = 0.
115     CONTINUE
1010    FORMAT(//,2X,'TERMINAL VALUE,',F10.3,' IS BEYOND LIMIT OF PREFEREN
      XCE CURVE'/' WHICH IS',F10.3,' AFTER TRANSFORMATION')
        RETURN
        END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR PREFR
 COMMON  1696  VARIABLES  12  PROGRAM  528

RELATIVE ENTRY POINT ADDRESS IS 004F (HEX)

END OF COMPILATION
```

```
PAGE 3      PREFR

// DUP

*DELETE               PREFR
CART ID 0021    DB ADDR 2C54    DB CNT 0027

*STORE         WS UA  PREFR
CART ID 0021    DB ADDR 34E8    DB CNT 0027


PAGE 1      PTREE

// JOB                                              PTREE

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021        0021         0000

V2 M08    ACTUAL 32K    CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**WRITE 3 TREE STAGES AND PRUNE R DE LANO 3/29/71
```

```
PAGE 2    PTREE    WRITE 3 TREE STAGES AND PRUNE R DE LANO 3/29/71
          SUBROUTINE PTREE(IA,YTOP,BIT,ISTG2,ISTG4,ICRCT,YAA,ISECT,ICTR)
          COMMON D(512),C(4,64), B(4,16),A(4,4)
C CALLED BY MAIN AND TREE
C IA RELATES TO FIRST STAGE BRANCHES AND NODES AT THEIR ENDS,IB SECOND,
C IC THIRD,ID FOURTH BRANCHES ONLY.
C ISTG2 AND ISTG4 ARE 1 WHEN TREE IS TO BE PRUNED. ISTG2 PRUNES STAGE 3.
C ISTG4 AND ICRCT DETERMINE PRUNING OF TERMINAL BRANCHES. SET IN MAIN AND INITB.
C ICRCT IS 1 DURING CORRECTION PHASE, CAUSING ZERO TERMINAL PROB TO BE PRINTED
          IPLT = 7
          IDSPL = 1
          HLFY = YTOP/2.
          YSTRT = HLFY—HLFY/BIT
          CALL FCHAR(0.,YSTRT,.1,.1,0.)
C WRITE O FOR DECISION NODE
          WRITE(IPLT,1000)
C B LOOP WRITES SECOND STAGE BRANCHES AND DECISION NODES
           DO 98 IB = 1,4
          YB = YTOP—IB*.64+.32—IA*2.56+2.56
          YBB = YB+HLFY/BIT
          IF(ISTG2)22,22,20
C HERE TO WRITE STAGE 2 PROB
20        IBB = IB+4*ISECT—4
          YP = YTOP—IB*.37—IA*2.56+2.07+HLFY/BIT
C HERE TO ASSIGN BRACKETS TO STAGE 2 PROB
          IBKT = IBB+72
          CALL GIOLB(IBKT)
          CALL FCHAR(3.0,YP,.1,.1,0.)
          WRITE(IPLT,1009)B(2,IBB)
          CALL GIORB
22        CALL FPLOT(1,2.05,YAA)
          CALL FPLOT(2,3.95,YBB)
          CALL FCHAR)3.95,YB,.1,.1,0.)
          WRITE(IPLT,1000)
C    C    LOOP WRITES THIRD STAGE BRANCHES AND CHANCE NODES
          DO 96 IC = 1,4
          YC = YTOP—IC*.16+.08—IB*.64+.64—IA*2.56+2.56
```

```
PAGE 3     PTREE     WRITE 3 TREE STAGES AND PRUNE R DE LANO 3/29/71
                YCC = YC+HLFY/BIT
C PRUNE STAGE 3
                IF(B(2,IBB))24,23,24
23              IF(ISTG2)24,24,30
24              CALL FPLOT(1,4.05,YBB)
                CALL FPLOT(2,5.95,YCC)
                CALL FCHAR(5.95,YC,.1,.1,0.)
                WRITE(IPLT,1001)
C D LOOP WRITES FOURTH STAGE BRANCHES
30              DO 94 ID = 1,4
                IYD = ID+4*IC+16*IB+64*IA—84
                YD = YTOP—.04*IYD
                YV = YD—.025
                IVD = ID+4*IC+16*IB+64*ISECT—84
                IPD = IVD+256
                IBKT = ID+4*IC+16*IB—13
C PRUNE STAGE 4
                IF(ISTG4)42,42,40
40              IF(ICRCT)400,400,402
400             IF(D(IPD)—.001)94,41,41
C HERE TO WRITE ZERO VALUES WHEN CORRECTING, ISTG4 = 1 —
C AND PROB IS GREATER THAN .001
C SO THEY MAY BE SELECTED FOR CORRECTION
402             CALL GIOLB(IBKT)
                CALL FCHAR(8.0,YV,.11,.11,0.)
                WRITE(IPLT,1009)D(IPD)
                CALL FCHAR(8.5,YV,.11,.11,0.)
                WRITE(IPLT,1007)D(IVD)
                CALL GIORB
                GO TO 42
C HERE TO SPREAD VALUES WHEN ISTG4 = 1, ICRCT = 0 AND PROB IS GREATER THAN .001
C HERE TO SPREAD VALUES WHEN ISTG = 1
41              ICTR = ICTR+1
                YD = YTOP—.25*ICTR
                YV = YD—.07
                CALL GIOLB(IBKT)
                CALL FCHAR(8.0,YV,.11,.11,0.)
```

```
PAGE 4    PTREE    WRITE 3 TREE STAGES AND PRUNE R DE LANO 3/29/71
                WRITE(IPLT,1009)D(IPD)
                CALL FCHAR(8.5,YV,.11,.11,0.)
                WRITE(IPLT,1007)D(IVD)
C HERE TO WRITE BRANCHES
 42             CALL FPLOT(1,6.05,YCC)
                CALL FPLOT(2,7.95,YD)
 94             CONTINUE
 96             CONTINUE
 98             CONTINUE
 1000           FORMAT('0')
 1001           FORMAT('0')
 1007           FORMAT(F10.2)
 1009           FORMAT(F5.3)
                RETURN
                END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR PTREE
 COMMON  1696  VARIABLES  34  PROGRAM  614

RELATIVE ENTRY POINT ADDRESS IS 0068 (HEX)

END OF COMPILATION

// DUP

*DELETE              PTREE
CART ID 0021   DB ADDR 2C54   DB CNT 002A

*STORE        WS UA PTREE
CART ID 0021   DB ADDR 34DF   DB CNT 002A
```

```
PAGE 1      SNSTY
                                                            SNSTY
// JOB
 LOG DRIVE    CART SPEC   CART AVAIL   PHY DRIVE
   0000          0021        0021        0000

V2 M08       ACTUAL 32K     CONFIG 32K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
**PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71
```

PAGE 2 SNSTY PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71

```
      SUBROUTINE SNSTY(IAMAX,YTOP,BIT,IFLAG,PI0,PI25,PI50,PI75,PI100,
     *PSHFT)
      REAL NEWAV
      INTEGER BRNCH(2,8)
      DIMENSION DEL(8), VAL(8)
      COMMON D(512),C(4,64), B(4,16),A(4,4)
C CALLED BY MAIN2
C DETERMINES PROB MOST SENSITIVE IN REDUCING VALUE TO NEXT LARGEST VALUE
C AND PROB MOST SENSITIVE IN PRODUCING A DECISION SHIFT
C IAMAX GIVES LOCATION OF MAX VALUE IN 'A' MATRIX. DETERMINED BY ACTS
C DEL 1—4 CONTAIN MIN DELTAS OF 'D'—BRANCH PROB THAT CHANGE DECISIONS,
C EITHER 'C' OR 'A'—BRANCH
C VAL 1—4 CONTAIN CORRESPONDING PROPOSITION VALUES
C BRANCH 2, 1—4 CONTAIN CORRESPONDING BRANCH NUMBERS
C DEL 5—8 CONTAIN DELTAS OF THE BRANCH PROB THAT PRODUCE THE LARGEST
C    RATIO OF (DELTA PROPOSITION VALUE)/(DELTA PROB THAT PRODUCED THE CHANGE)
C VAL 5—8 CONTAIN CORRESPONDING PROPOSITION VALUES
C BRANCH 2, 5—8 CONTAIN CORRESPONDING BRANCH NUMBERS
C BRANCH 1,X IS 1 FOR B BRANCH NUMBER, 0 FOR D BRANCH NUMBER
C IBFLG IS ZERO FOR 'D'—BRANCH, ONE FOR 'B'BRANCH. SET HERE
      IDSPL = 1
      IPRNT = 5
C ZERO THIS PROGRAMS ARRAYS
      DO 1 M = 1,8
      BRNCH(1,M) = 0
      BRNCH(2,M) = 0
      DEL(M) = 0.
      VAL(M) = 0.
1     CONTINUE
C     LOOK FOR VALUE (ANEXT) CLOSEST TO LARGEST (A(3,IAMAX))
      IANXT = 0
      ANEXT = A(3,1)/2.
      DO 10 I = 1,4
      IF(A(3,I)—A(3,IAMAX))3,10,10
3     IF(ANEXT—A(3,I))5,10,10
5     ANEXT = A(3,I)
      IANXT = I
```

```
PAGE 3    SNSTY    PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71

10        CONTINUE
C IANXT AND ANEXT REMAIN ZERO IF NODE 'A' HAS ONE OR NO ACTIVE BRANCHES —
C I.E. NO DECISION, HENCE NEGLECTED
C DETERMINE LARGEST AND SMALLEST 'B' BRANCH VALUES IN DECISION PATH
          ISTRT = 4*IAMAX—3
          IFIN = ISTRT+3
          BMAX = B(3,ISTRT)
          BMIN = BMAX
          DO 20 I = ISTRT,IFIN
          IF(B(2,I))20,20,14
14        IF(BMAX—B(3,I))16,18,18
16        BMAX = B(3,I)
          IBMAX = I
18        IF(B(3,I)—BMIN)19,20,20
19        BMIN = B(3,I)
          IBMIN = I
20        CONTINUE
C DELPB IS REDUCTION IN PROB OF MAX VALUE 'B'—BRANCH THAT WILL SHIFT DECISION
C TO NEXT MOST VALUABLE 'A'BRANCH. AS PROB(B—MAX) IS REDUCED IT IS ADDED TO
C PROB(B—MIN)
          DELPB = (ANEXT—A(3,IAMAX))/(BMIN—BMAX)
C DETERMINE LARGEST AND SMALLEST TERMINAL VALUES
          TVMAX = D(1)
          TVMIN = TVMAX
          DO 40 I = 1,256
C SKIP WHEN PROB IS ZERO
          IF(D(I+256))40,40,32
32        IF(TVMAX—D(I))33,35,35
33        TVMAX = D(I)
35        IF(D(I)—TVMIN)37,40,40
37        TVMIN = D(I)
40        CONTINUE
C LOOK FOR MAX VALUE (CMAX) THEN VALUE (CNEXT) CLOSEST TO MAX VALUE
C MOST SENSITIVE TERMINAL PROB WILL BE THE ONE WHICH CHANGES THE LEAST IN
C CAUSING A CHANGE IN A 'C' DECISION
C J IS B ARRAY COUNTER
C L AND K ARE THE ARRAY COUNTERS FOR THE STORAGE ARRAYS— BRNCH, DEL AND VAL
```

PAGE 4 SNSTY PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71

```
      L = 0
      ISTRJ = 4*IAMAX—3
      IFINJ = ISTRJ+3
      DO 100 J = ISTRJ,IFINJ
      L = L+1
C SKIP IF B BRANCH PROB IS ZERO
      IF(B(2,J))100,100,41
41    ISTRI = 4*J—3
      IFINI = ISTRI+3
      CMAX = TVMIN
      DO 46 I = ISTRI,IFINI
C ELIMINATE UNDESIGNATED ZEROS
      IF(C(1,I))44,46,44
44    IF(C(3,I)—CMAX)46,46,45
45    CMAX = C(3,I)
      ICMAX = I
46    CONTINUE
C INITIATE CNEXT
      CNEXT = CMAX
      DO 49 I = ISTRI,IFINI
      IF(I—ICMAX)47,49,47
47    IF(C(1,I))48,49,48
48    CNEXT = C(3,I)
      ICNXT = I
49    CONTINUE
      DO 60 I = ISTRI,IFINI
C ELIMINATE UNDESIGNATED ZEROS
      IF(C(1,I))60,60,52
52    IF(C(3,I)—CMAX)53,60,60
53    IF(C(3,I)—CNEXT)60,60,55
55    CNEXT = C(3,1)
      ICNXT = I
60    CONTINUE
C DETERMINE LARGEST AND SMALLEST 'D'—BRANCHES FOR CMAX
      ISTRT = 4*ICMAX—3
      IFINI = ISTRT+3
      DMAX = TVMIN
```

```
PAGE 5      SNSTY      PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71
                  DMIN = TVMAX
                  DO 70 I = ISTRT,IFINI
C SKIP TERMINAL NODES HAVING PROB OF ZERO OR ONE
                  IF(D(I+256))70,70,600
600               IF(1—D(I+256))100,100,64
64                IF(D(I)—DMAX)67,67,65
65                DMAX = D(I)
                  IDMAX = I
67                IF(DMIN—D(I))70,70,69
69                DMIN = D(I)
70                CONTINUE
C DELPD IS REDUCTION IN PROB OF MAX VALUE 'D'—BRANCH THAT WILL SHIFT DECISION
C TO NEXT MOST VALUABLE 'C'BRANCH. AS PROB(D—MAX) IS REDUCED IT IS ADDED TO
C PROB(D—MIN)
                  DELPD = (CNEXT—CMAX)/(DMIN—DMAX)
C DETERMINE REDUCTION IN VALUE CAUSED BY CHANGE OF DELPD CHANGING C DECISION
                  NEWAV = A(3,IAMAX)—(CMAX—CNEXT)*B(2,J)
C DEPD2 IS REDUCTION IN PROB OF MAX VALUE 'D'—BRANCH THAT WILL SHIFT DECISION
C TO NEXT MOST VALUABLE 'A'BRANCH. AS PROB(D—MAX) IS REDUCED IT IS ADDED TO
C PROB(D—MIN)
                  DLPD2 = (A(3,IAMAX)—ANEXT)/(DMAX—DMIN)/B(2,J)
C STORE SMALLER DELTA 'D'—BRANCH PROB
                  IF(DLPD2—DELPD)72,72,71
71                DEL(L) = DELPD
                  VAL(L) = NEWAV
                  GO TO 73
72                DEL(L) = DLPD2
                  VAL(L) = ANEXT
C DETERMINE LARGEST RATIO OF (DELTA VALUE)/(DELTA D PROB)
73                DLVD1 = (A(3,IAMAX)—NEWAV)/DELPD
                  DLVD2 = (A(3,IAMAX)—ANEXT)/DLPD2
                  IF(DLVD1—DLVD2)75,75,74
74                DELVD = DLVD1
                  DELP = DELPD
                  VSAVE = NEWAV
                  GO TO 77
75                DELVD = DLVD2
```

```
PAGE 6      SNSTY      PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71
                 VSAVE = ANEXT
                 DELP = DLPD2
77               DELVB = (A(3,IAMAX)—ANEXT)/DELPB
                 BRNCH(2,L) = IDMAX
C PICK LARGEST RATIO (DELTA VALUE)/(DELTA PROB). IT WILL BE FOR EITHER
C 'D'—BRANCH OR 'B'—BRANCH
                 IF(DELVD—DELVB)81,79,79
79               REDVL = VSAVE
                 IBRCH = IDMAX
                 WRITE(IPRNT,1068)TVMAX,TVMIN,IBRCH,DELP,REDVL
                 BRNCH(1,L+4) = 0
                 GO TO 90
81               REDVL = ANEXT
                 IBRCH = IBMAX
                 DELP = DELPB
                 WRITE(IPRINT,1070)TVMAX,TVMIN,IBRCH,DELP,REDVL
                 BRNCH(1,L+4) = 1
C STORE IN ARRAYS
90               BRNCH(2,L+4) = IBRCH
                 DEL(L+4) = DELP
                 VAL(L+4) = REDVL
100              CONTINUE
C INVERT UTILES
                 IF(IFLAG)102,102,101
101              V = TVMAX
                 CALL       VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
                 TVMAX = V+PSHFT
                 V = TVMIN
                 CALL       VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
                 TVMIN = V+PSHFT
C
C DETERMINE PROB MOST SENSITIVE IN PRODUCING A DECISION CHANGE
C
102              PMIND = 1.
                 DO 110 K = 1,4
C SKIP WHEN PROB IS ZERO
                 IF(DEL(K))110,110,103
```

```
PAGE 7     SNSTY     PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71

103          IF(DEL(K)—PMIND)105,105,110
105          PMIND = DEL(K)
             ISAVE = K
110          CONTINUE
             IF(DELPB—PMIND)115,115,113
C HERE IF DELTA P(D) IS MOST SENSITIVE IN CHANGING DECISION
C   IWRTE DETERMINES DISPLAY MESSAGE SELECTION
113          DELP = PMIND
             IWRTE = 1
             IBFLG = 0
             IBRCH = BRNCH(2,ISAVE)
             REDVL = VAL(ISAVE)
             IF(REDVL—ANEXT)114,1130,114
1130         IF(IFLAG)1132,1132,1131
1131         V = REDVL
             CALL      VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
             REDVL = V+PSHFT
1132         WRITE(IDSPL,1075)IBRCH,DELP,D(IBRCH+256),IANXT,REDVL
             WRITE(IPRNT,1075)IBRCH,DELP,D(IBRCH+256),IANXT,REDVL
             GO TO 116
114          IF(IFLAG)1141,1141,1140
1140         V = REDVL
             CALL      VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
             REDVL = V+PSHFT
1141         WRITE(IDSPL,1071)IBRCH,DELP,D(IBRCH+256),ICNXT,REDVL
             WRITE(IPRNT,1071)IBRCH,DELP,D(IBRCH+256),ICNXT,REDVL
             GO TO 116
C HERE IF DELTA P(B) IS MOST SENSITIVE IN CHANGING DECISION
115          DELP = DELPB
             IWRTE = 2
             IBFLG = 1
             IBRCH = IBMAX
             REDVL = ANEXT
             IF(IFLAG)1151,1151,1150
1150         V = REDVL
             CALL      VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
```

```
PAGE 8      SNSTY      PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71

                 REDVL = V+PSHFT
1151             WRITE(IPRINT,1072)IBRCH,DELP,B(2,IBMAX),IANXT,REDVL
                 WRITE(IDSPL,1072)IBRCH,DELP,B(2,IBMAX),IANXT,REDVL
116              CALL      SPRED(YTOP,BIT,IBRCH,IBFLG)
C
C DETERMINE PROB MOST SENSITIVE IN PRODUCING A VALUE CHANGE
C
                 DVRMX = 0.
                 DO 120 K = 5,8
C SKIP ZEROS
                 IF(DEL(K))117,120,117
117              DVR = (A(3,IAMAX)—VAL(K))/DEL(K)
                 IF(DVRMX—DVR)120,120,118
118              DVRMX = DVR
                 ISAVE = K
120              CONTINUE
                 IF(BRNCH(1,ISAVE))121,121,124
C HERE IF 'D' BRANCH PROB MOST SENSITIVE IN PRODUCING VALUE CHANGE
C SKIP DISPLAY IF NOT A NEW BRANCH
121              IF(IBRCH—BRNCH(2,ISAVE))122,123,122
122              IWRTE = 3
123              IBRCH = BRNCH(2,ISAVE)
                 IBFLG = 0
                 DELP = DEL(ISAVE)
                 REDVL = VAL(ISAVE)
                 WRITE(IPRINT,1067)TVMAX,TVMIN,IBRCH,DELP,REDVL
                 GO TO 150
C HERE IF 'B' BRANCH PROB MOST SENSITIVE IN PRODUCING VALUE CHANGE
C SKIP DISPLAY IF NOT A NEW BRANCH
124              IF(IBRCH—IBMAX)125,126,125
125              IWRTE = 4
126              IBRCH = IBMAX
                 IBFLG = 1
                 DELP = DELPB
                 REDVL = ANEXT
                 WRITE(IPRNT,1069)TVMAX,TVMIN,IBRCH,DELP,REDVL
```

```
PAGE 9    SNSTY    PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71

150         CONTINUE
C SELECT DISPLAY MESSAGE
            GO TO(155,155,160,165),IWRTE
155         WRITE(IDSPL,1073)
            WRITE(IPRNT,1073)
            GO TO 170
160         IF(IFLAG)162,162,161
161         V = REDVL
            CALL      VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
            REDVL = V+PSHFT
162         WRITE(IDSPL,1067)TVMAX,TVMIN,IBRCH,DELP,REDVL
            GO TO 167
165         IF(IFLAG)166,166,1651
1651        V = REDVL
            CALL      VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
            REDVL = V+PSHFT
166         WRITE(IDSPL,1069)TVMAX,TVMIN,IBRCH,DELP,REDVL
167         CALL      SPRED(YTOP,BIT,IBRCH,IBFLG)
1067        FORMAT('YOU SAID YOUR MAX GAIN WAS',F10.2,'AND YOUR MIN WAS',F10
           *.2/
           *' IF PROB OF BRANCH ',13,' DECREASES BY',F6.3,' YOUR GAIN/LOSS IS
           *',F10.2)
1068        FORMAT(' YOU TOLD ME YOUR MAXIMUM GAIN IS' ,F10.2,/
           *' AND YOUR MINIMUM GAIN IS' ,F10.2,/
           *' IF THE PROB OF BRANCH' ,13,' DECREASES BY' ,F6.3,/
           *' YOUR GAIN/LOSS BECOMES' ,F10.2)
1069        FORMAT(' YOU SAID YOUR MAX GAIN WAS',F10.2,' AND YOUR MIN WAS'F10
           *.2/
           *' IF PROB OF BRANCH B',13,' DECREASES BY'F6.3,' YOUR GAIN/LOSS IS
           *',F10.2)
1070        FORMAT(' YOU TOLD ME YOUR MAXIMUM GAIN IS' ,F10.2,/
           *' AND YOUR MINIMUM GAIN IS' ,F10.2,/
           *'IF THE PROB OF BRANCH B',13,'DECREASES BY',F6.3,/
           *' YOUR GAIN/LOSS BECOMES' ,F10.2)
1070        FORMAT(' IF THE PROB OF BRANCH ',13,' DECREASES BY',F6.3,' FROM',F
           *6.3,/
           *' YOUR SECOND DECISION CHANGES TO BRANCH C',13,'. YOUR VALUE IS',F
```

```
PAGE 10    SNSTY    PERFORMS SENSITIVITY ANAYYSIS — R DE LANO 04/17/71 REV 06/18/71

                *10.2)
1072             FORMAT(' IF THE PROB OF BRANCH B',13,' DECREASES BY',F6.3,' FROM',
                *F6.3,/
                *' YOUR FIRST DECISION CHANGES TO A',13,' WITH A VALUE OF',F10.2)
1073             FORMAT(' THE SAME PROB IS ALSO MOST SENSITIVE IN CHANGING VALUE',/
                *)
1075             FORMAT(' IF THE PROB OF BRANCH ',13,' DECREASES BY',F6.3,' FROM',F
                *6.3,/
                *' YOUR FIRST DECISION CHANGES TO BRANCH A',13,'. YOUR VALUE IS',F
                *10.2)
170              RETURN
                 END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR SNSTY
 COMMON  1696  VARIABLES  124  PROGRAM  2112

RELATIVE ENTRY POINT ADDRESS IS 0271 (HEX)

END OF COMPILATION

// DUP

*DELETE              SNSTY
CART ID 0021   DB ADDR 2C54   DB CNT 008E

*STORE        WS  UA  SNSTY
CART ID 0021   DB ADDR 3479   DB CNT 008E
```

```
PAGE 1      SPRED

// JOB                                                  SPRED

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021         0021        0000

V2 M08     ACTUAL 32K     CONFIG 32K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
**BROADEN PROB BRANCHES— R DE LANO 04/26/71
```

```
PAGE 2    SPRED    BROADEN PROB BRANCHES— R DE LANO 04/26/71
          SUBROUTINE SPRED(YTOP,BIT,IBRCH,IBFLG)
          COMMON D(512),C(4,64), B(4,16),A(4,4)
C CALLED BY SNSTY
C BROADENS BRANCHES HAVING MOST SENSITIVE PROB
C IBFLG IS ONE WHEN IBRCH IS A 'B'—BRANCH. SET IN SNSTY
          IPLT = 7
          HLFY = YTOP/2.
          IF(IBFLG)15,15,5
C HERE TO BROADEN A 'B'—BRANCH
5         IA = IBRCH/4+1
          YA = YTOP—IA*2.56+1.28
          YB = YTOP—IBRCH*.64+.32+HLFY/BIT
          DO 10 ISPRD = 1,4
          YBB = YB—ISPRD*.015+.04
          CALL FPLOT(1,2.05,YA)
          CALL FPLOT(2,3.95,YBB)
10        CONTINUE
          GO TO 40
C HERE TO BROADEN A 'D'—BRANCH
15        IC = IBRCH/4+1
          YC = YTOP—IC*.16+.08+HLFY/BIT
C SINCE TERMINAL BRANCHES ARE PRUNED, DETERMINE N—THE NUMBER OF
C PREVIOUS BRANCHES
          N = 0
          DO 20 J = 1,IBRCH
          IF(D(J+256)—.001)20,20,17
17        N = N+1
20        CONTINUE
          YD = YTOP—.25*N
          DO 30 ISPRD = 1,4
          YDD = YD—ISPRD*.015+.04
          CALL FPLOT(1,6.05,YC)
          CALL FPLOT(2,7.95,YDD)
30        CONTINUE
40        RETURN
          END
```

```
PAGE   3   SPRED   BROADEN   PROB   BRANCHES—   R   DE   LANO   04/26/71

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR SPRED
 COMMON  1696  VARIABLES  24  PROGRAM  252

RELATIVE ENTRY POINT ADDRESS IS 003B (HEX)

END OF COMPILATION

// DUP

*DELETE              SPRED
CART ID 0021     DB ADDR 2C54     DB CNT 0012

*STORE        WS UA SPRED
CART ID 0021     DB ADDR 34F7     DB CNT 0012



PAGE 1     TREE

// JOB                                                  TREE

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021         0021         0000

V2 M08     ACTUAL 32K     CONFIG 32K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
**DISPLAY STAGE ONE — R DE LANO 2/28/71 REV 3/14/71
```

```
PAGE 2    TREE    DISPLAY STAGE ONE — R DE LANO 2/28/71 REV 3/14/71
          SUBROUTINE TREE(ISTG2,ISTG4,YTOP,BIT,ICRCT)
          COMMON D(512),C(4,64), B(4,16),A(4,4)
C IA RELATES TO FIRST STAGE BRANCHES AND NODES AT THEIR ENDS,IB SECOND,
C IC THIRD,ID FOURTH BRANCHES ONLY.
C ISTG2 AND ISTG4 ARE 1 WHEN TREE IS TO BE PRUNED AND THEREAFTER
C ICRCT IS 1 DURING CORRECTION PHASE, CAUSING ZERO TERMINAL PROB TO BE PRINTED
          IPLT = 7
          IDSPL = 1
          TWBIT = 110.
          HLFY = YTOP/2.
          ICTR = 0.
C OUTSIDE LOOP WRITES FIRST FOUR BRANCHES
          DO 100 IA = 1,4
          ISECT = IA
          YA = YTOP—IA*2.56+1.28
          CALL FPLOT(1,.1,HLFY)
          CALL FPLOT(2,1.95,YA)
C WRITE 0 FOR CHANCE NODE
          YA = YA—HLFY/TWBIT
          CALL FCHAR (1.95,YA,.1,.1,0.)
          WRITE (IPLT,1001)
          YAA = YA+HLFY/BIT
          CALL      PTREE(IA,YTOP,BIT,ISTG2,ISTG4,ICRCT,YAA,ISECT,ICTR)
100       CONTINUE
1001      FORMAT('O')
          RETURN
          END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR TREE
 COMMON 1696   VARIABLES  14   PROGRAM  138

RELATIVE ENTRY POINT ADDRESS IS 0024 (HEX)

END OF COMPILATION
```

```
PAGE 3     TREE

// DUP

*DELETE                    TREE
CART ID 0021    DB ADDR 2C54    DB CNT 000A

*STORE        WS  UA  TREE
CART ID 0021    DB ADDR 3509    DB CNT 000A


PAGE 1     VINVT

// JOB                                          VINVT

 LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
   0000        0021        0021         0000

V2 M08    ACTUAL 32K    CONFIG 32K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
**INVERT V— R DE LANO 06/18/71
```

```
PAGE 2      VINVT      INVERT V— R DE LANO 06/18/71

            SUBROUTINE VINVT(V,PI0,PI25,PI50,PI75,PI100,PSHFT)
C CALLED SEVEN TIMES BY SNSTY, ONCE BY MAIN2
C TRANSFORMS V INTO SAME DIMENSION AS TERMINAL VALUES
C USES LINEAR INTERPOLATION
            IF(V—.25)4,4,6
4           V = V*(PI25—PI0)/.25+PI0
            GO TO 19
6           IF(V—.50)8,8,10
8           V = V*(PI50—PI25)/.25—PI50+2*PI25
            GO TO 19
10          IF(V—.75)12,12,14
12          V = V*(PI75—PI50)/.25—2*PI75+3*PI50
            GO TO 19
14          V = V*(PI100—PI75)/.25—3*PI100+4*PI175
19          RETURN
            END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR VINVT
 COMMON  0  VARIABLES  4  PROGRAM  164

RELATIVE ENTRY POINT ADDRESS IS 000D (HEX)

END OF COMPILATION

// DUP

*DELETE              VINVT
CART ID 0021     DB ADDR 2C54     DB CNT 000D

*STORE        WS  UA  VINVT
CART ID 0021     DB ADDR 3503     DB CNT 000D
```

WHAT WE CLAIM IS:—

1. A data processing system including a central processor, a graphic display unit, and programming means which cause the system to generate and display a decision tree, to assign probabilities and expected values to selected branches of the tree in response to input data, to modify the tree in response to operator selection of branches therein, to calculate modified probabilities and expected values for selected branches of the modified tree, and to display the modified tree and modified probabilities and expected values on the display unit.

2. A data processing system as claimed in claim 1 wherein the system is operable to determine highest expected value tree paths and to indicate such paths on the displayed tree diagrams.

3. A data processing system as claimed in claim 1 or claim 2 wherein the system is operable to determine, from a set of branches extending from a node, that branch which is most sensitive to a change in the highest expected value tree path in response to a change in the probability assigned to the branch.

4. A data processing system as claimed in claim 3, wherein the system is operable to calculate the magnitude of the change in probability assigned to each branch which becomes part of the highest expected value tree path, and to display an indication of said magnitude.

5. A data processing system as claimed in any of the previous claims in which the system is operable to detect and display errors made in the assignment of said probabilities and values.

6. A data processing system as claimed in claim 5 wherein the system is operable to determine whether or not the sum of the probabilities assigned to a set of branches extending from a node is unity.

7. A data processing system as claimed in any of the previous claims, said system being operable to calculate and display preference curves of utiles plotted against values and to calculate the resulting expectations in utiles at selected nodes of the trees.

8. A data processing system substantially as described herein with reference to Figure 21 when programmed as described herein with reference to Figures 8 and 9 of the accompanying drawings.

9. A method of programming a computer substantially as described herein with reference to Figures 8 to 20 of the accompanying drawings.

A. G. F. HAWKINS,
Chartered Patent Agent,
Agent for the Applicants.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1975.
Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

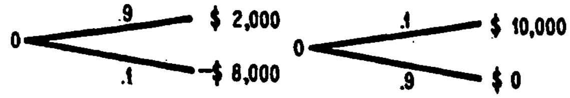



FIG. 1

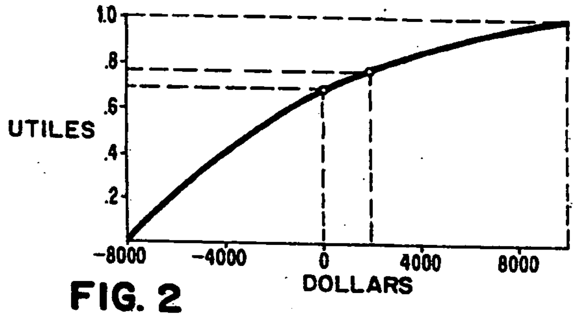



FIG. 2

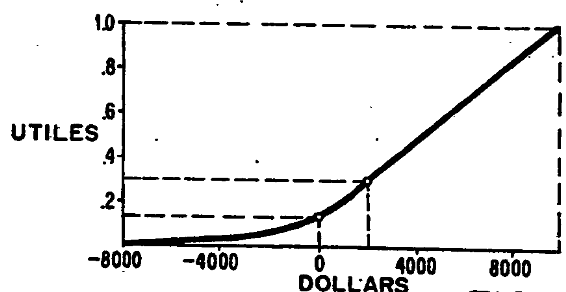



FIG. 3

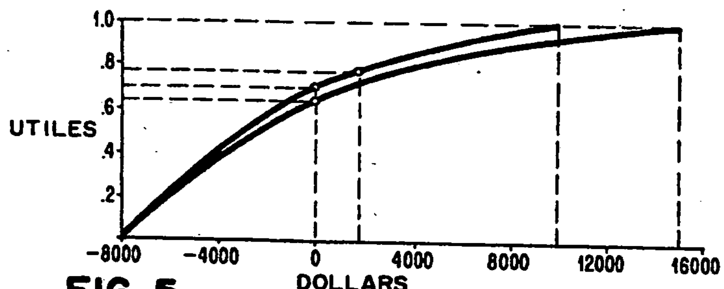



FIG. 5

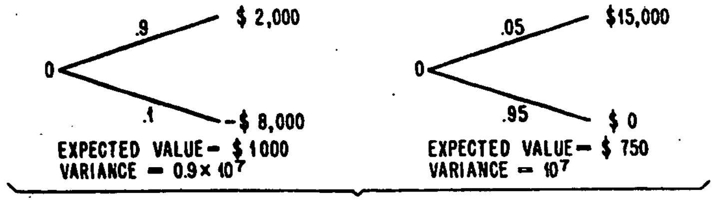



FIG. 4

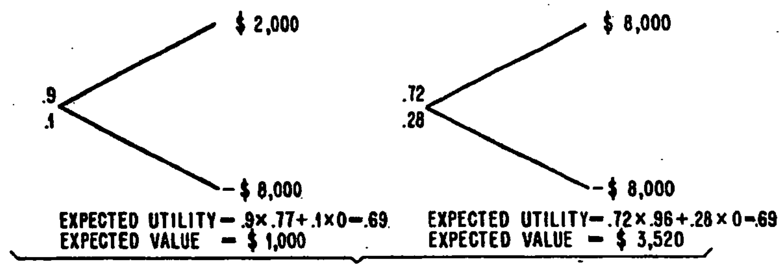



FIG. 6

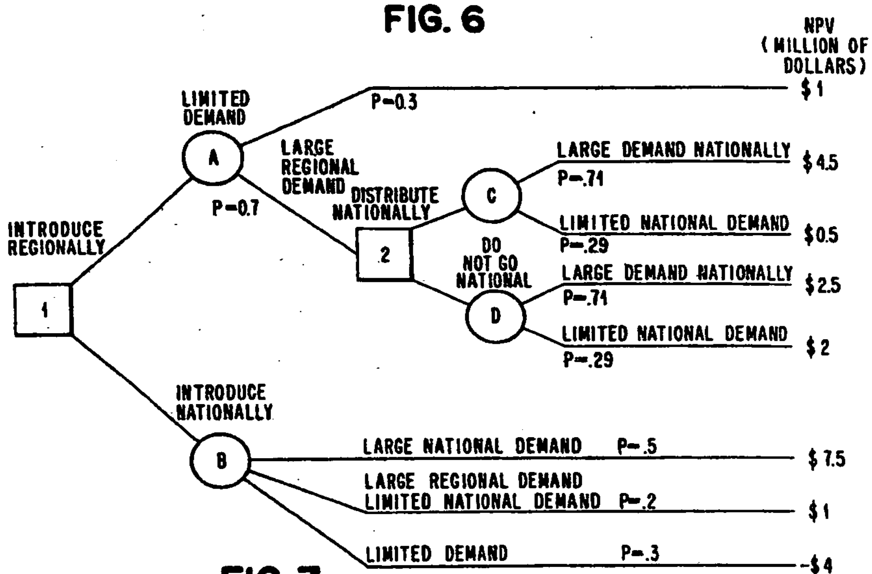
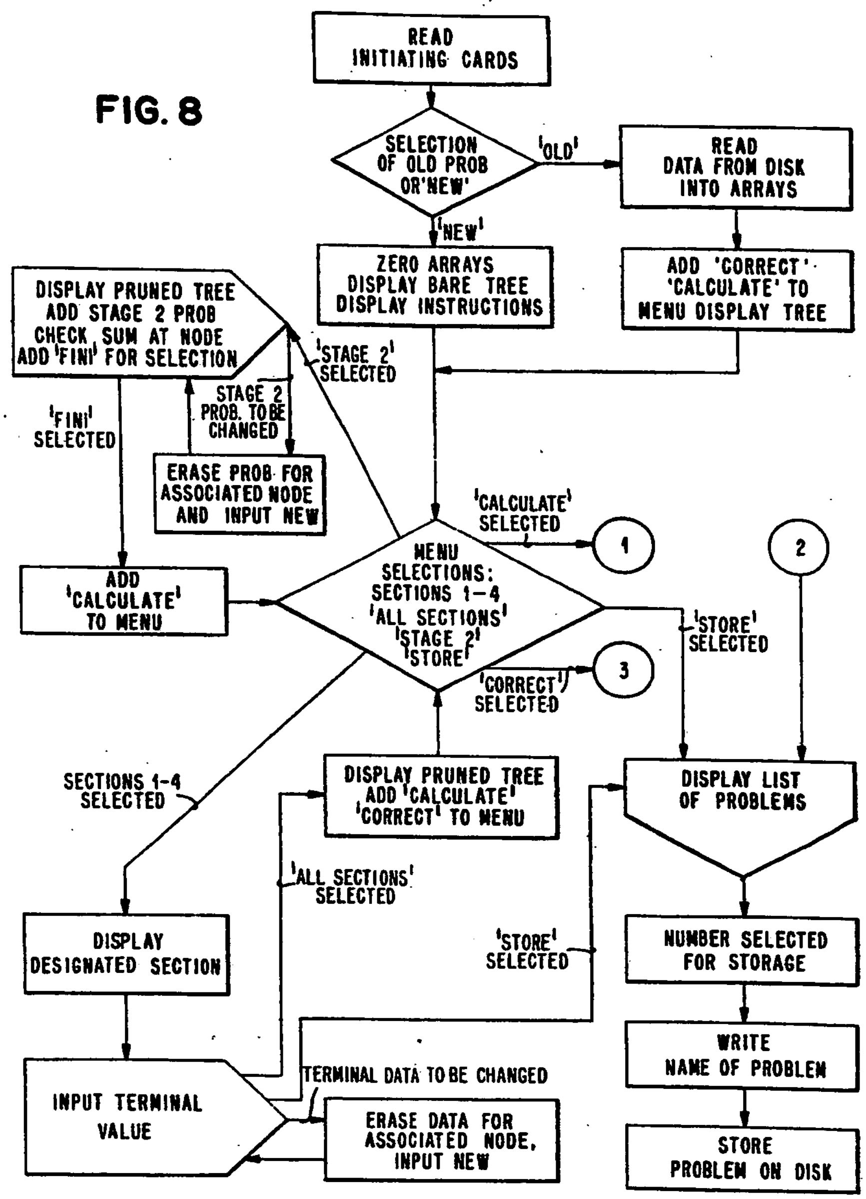



FIG. 7

FIG. 8


READ INITIATING CARDS
SELECTION OF OLD PROB OR 'NEW'
'OLD'
READ DATA FROM DISK INTO ARRAYS
'NEW'
ZERO ARRAYS DISPLAY BARE TREE DISPLAY INSTRUCTIONS
ADD 'CORRECT' 'CALCULATE' TO MENU DISPLAY TREE
DISPLAY PRUNED TREE ADD STAGE 2 PROB CHECK SUM AT NODE ADD 'FINI' FOR SELECTION
'STAGE 2' SELECTED
STAGE 2 PROB. TO BE CHANGED
'FINI' SELECTED
ERASE PROB FOR ASSOCIATED NODE AND INPUT NEW
'CALCULATE' SELECTED
1
2
MENU SELECTIONS: SECTIONS 1-4 'ALL SECTIONS' 'STAGE 2' 'STORE'
ADD 'CALCULATE' TO MENU
'STORE' SELECTED
'CORRECT' SELECTED
3
SECTIONS 1-4 SELECTED
DISPLAY PRUNED TREE ADD 'CALCULATE' 'CORRECT' TO MENU
DISPLAY LIST OF PROBLEMS
'ALL SECTIONS' SELECTED
DISPLAY DESIGNATED SECTION
'STORE' SELECTED
NUMBER SELECTED FOR STORAGE
WRITE NAME OF PROBLEM
INPUT TERMINAL VALUE
TERMINAL DATA TO BE CHANGED
ERASE DATA FOR ASSOCIATED NODE, INPUT NEW
STORE PROBLEM ON DISK

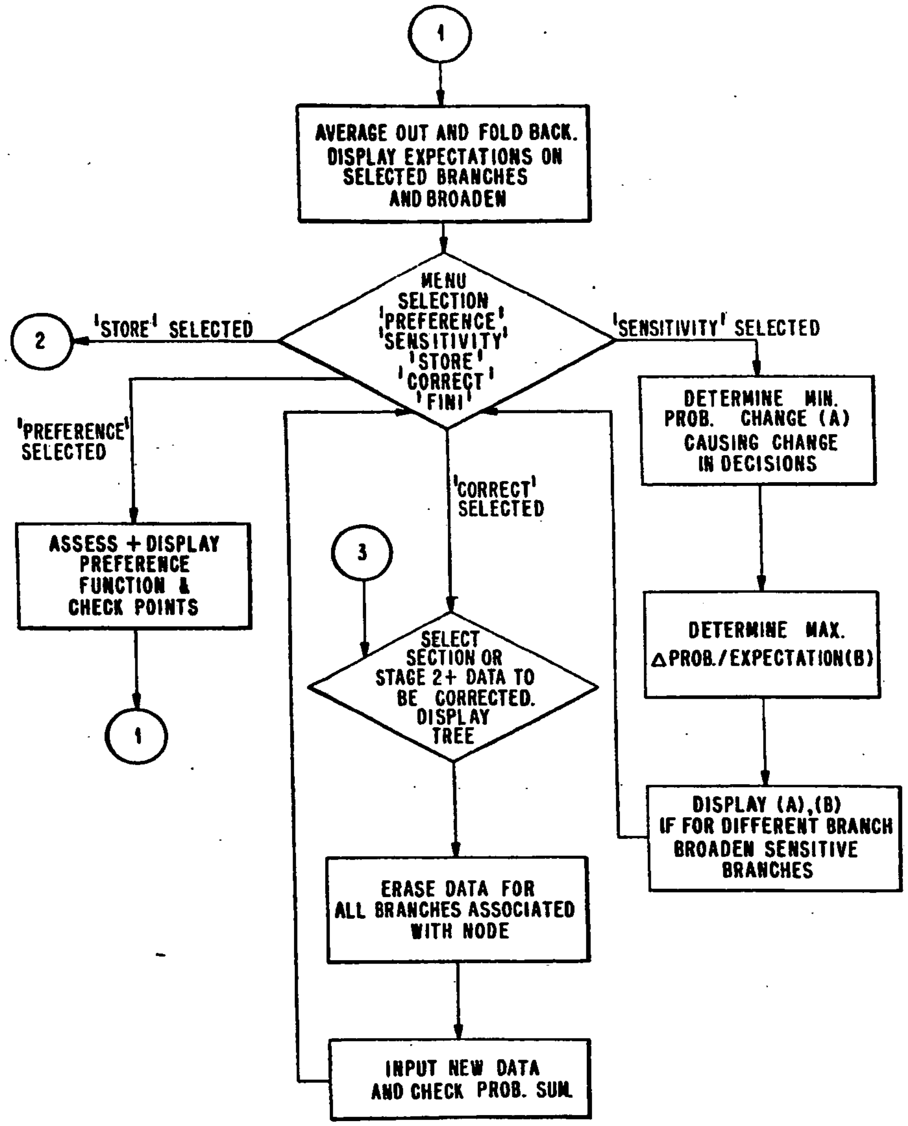

1
AVERAGE OUT AND FOLD BACK. DISPLAY EXPECTATIONS ON SELECTED BRANCHES AND BROADEN
MENU SELECTION 'PREFERENCE' 'SENSITIVITY' 'STORE' 'CORRECT' 'FINI'
2
'STORE' SELECTED
'SENSITIVITY' SELECTED
'PREFERENCE' SELECTED
'CORRECT' SELECTED
ASSESS + DISPLAY PREFERENCE FUNCTION & CHECK POINTS
1
3
SELECT SECTION OR STAGE 2+ DATA TO BE CORRECTED. DISPLAY TREE
DETERMINE MIN. PROB. CHANGE (A) CAUSING CHANGE IN DECISIONS
DETERMINE MAX. ΔPROB./EXPECTATION(B)
DISPLAY (A),(B) IF FOR DIFFERENT BRANCH BROADEN SENSITIVE BRANCHES
ERASE DATA FOR ALL BRANCHES ASSOCIATED WITH NODE
INPUT NEW DATA AND CHECK PROB. SUM.

FIG. 9

PREFERENCE
SENSITIVITY
STORE
CORRECT
FINI

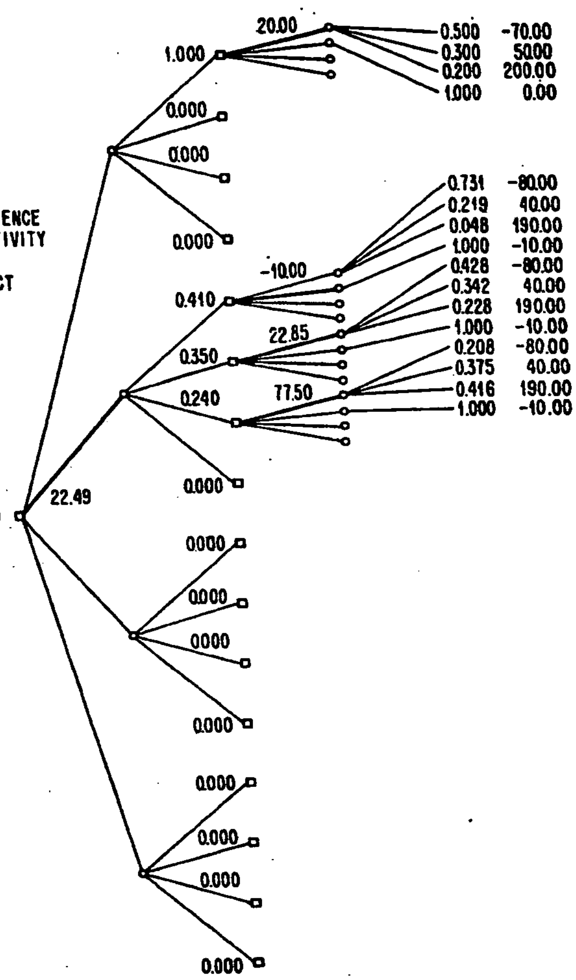
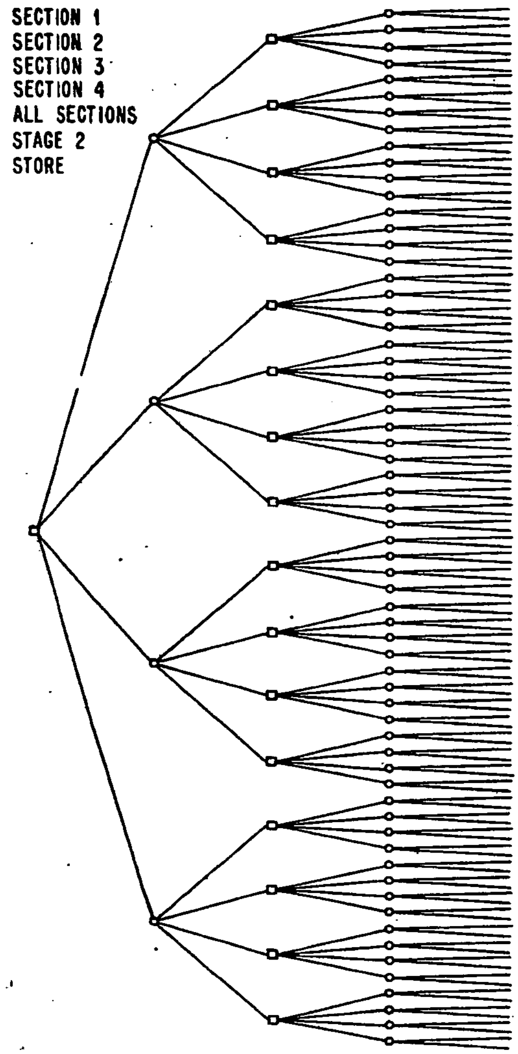



**FIG.10**

I HAVE COMPUTED THE VALUE OF YOUR PROPOSITION
THE MOST VALUABLE CHOICES ARE PRINTED AND SELECTED BRANCHES ARE HEAVY
SELECT PREFERENCE TO DO A UTILITY ANALYSIS
SENSITIVITY FOR SENSITIVITY ANALYSIS, FINI WHEN FINISHED, PLEASE SELECT

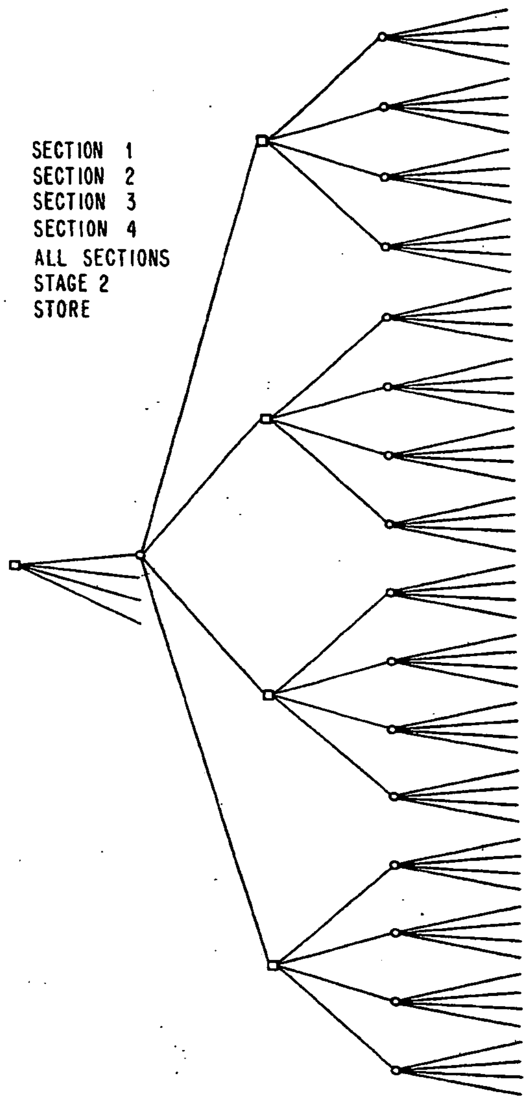
1.. RAIFFA DOLLARS
2.. PROB ASSGN ERRORS
3..
4..
5..
6..
7..
8..
9..
10..
11..
12..
13..
14..
15..
16..
.. NEW
SELECT DESIRED PROBLEM OR 'NEW' WITH LIGHT PEN
WAITING FOR YOUR SELECTION

**FIG.11**




THIS IS YOUR INITIAL DECISION TREE
THERE ARE FOUR EQUAL SECTIONS OF THE TREE FROM TOP TO BOTTOM
USE LIGHT PEN TO SELECT A SECTION TO BE MAGNIFIED
WAITING FOR YOU TO MAKE A SELECTION

FIG. 12




TYPE TERMINAL VALUE FOR BRANCH 1 IF VALUE WOULD EXCEED 3 DIGITS
USE K AFTER DIGITS FOR TIMES 1.000, M FOR TIMES 1.000.000
THEN PUSH ALTN CODING AND END AT THE SAME TIME

**FIG. 13**

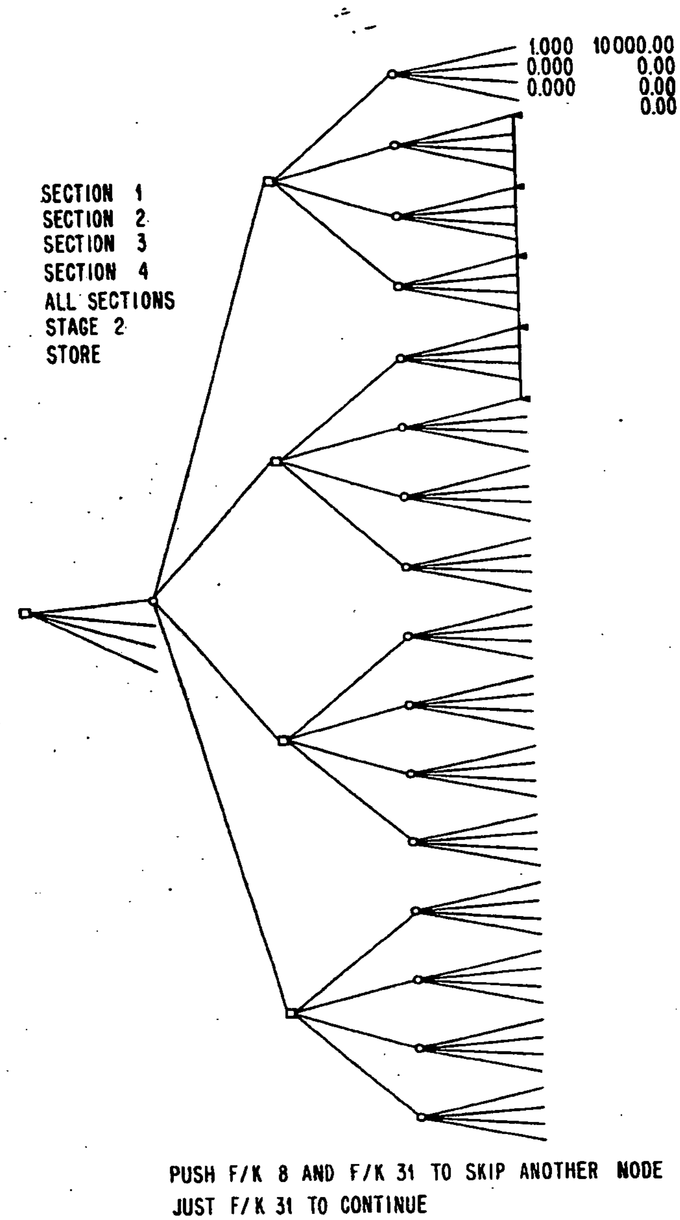

1.000 10000.00
0.000 0.00
0.000 0.00
0.00
SECTION 1
SECTION 2
SECTION 3
SECTION 4
ALL SECTIONS
STAGE 2
STORE
PUSH F/K 8 AND F/K 31 TO SKIP ANOTHER NODE
JUST F/K 31 TO CONTINUE

FIG.14

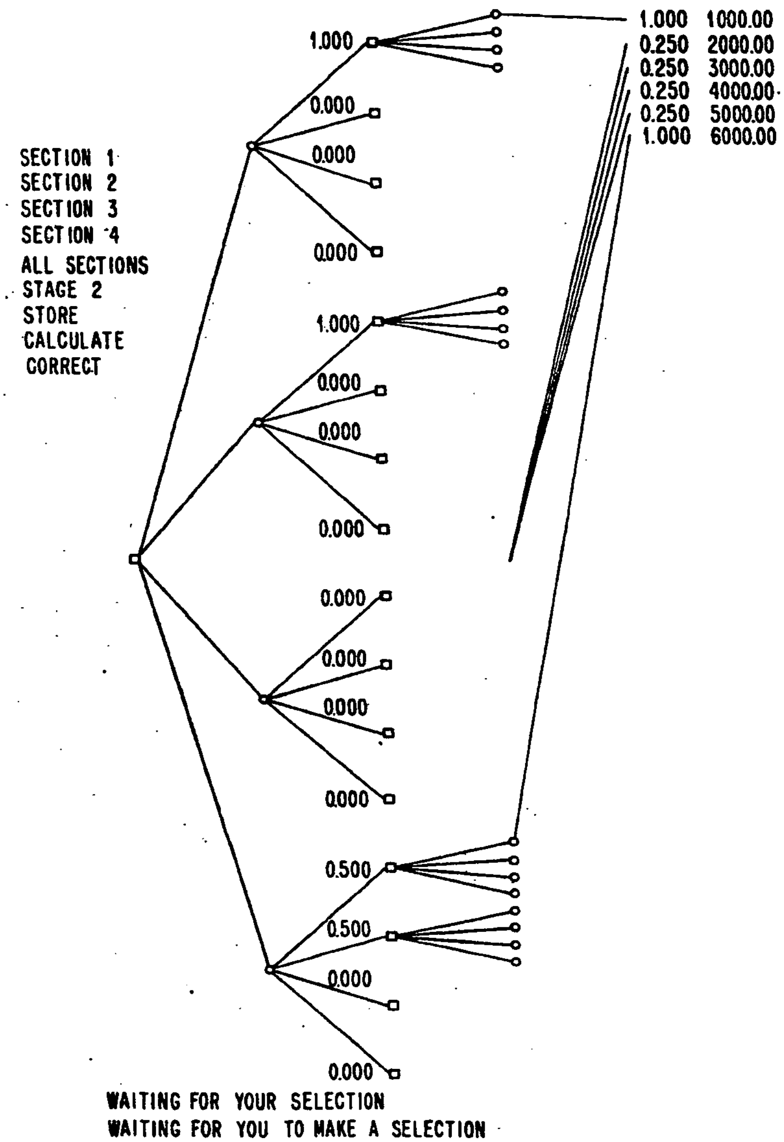

1.000 1000.00
0.250 2000.00
0.250 3000.00
0.250 4000.00
0.250 5000.00
1.000 6000.00
1.000
0.000
0.000
0.000
SECTION 1
SECTION 2
SECTION 3
SECTION 4
ALL SECTIONS
STAGE 2
STORE
CALCULATE
CORRECT
1.000
0.000
0.000
0.000
0.000
0.000
0.000
0.000
0.500
0.500
0.000
0.000
WAITING FOR YOUR SELECTION
WAITING FOR YOU TO MAKE A SELECTION

FIG. 15

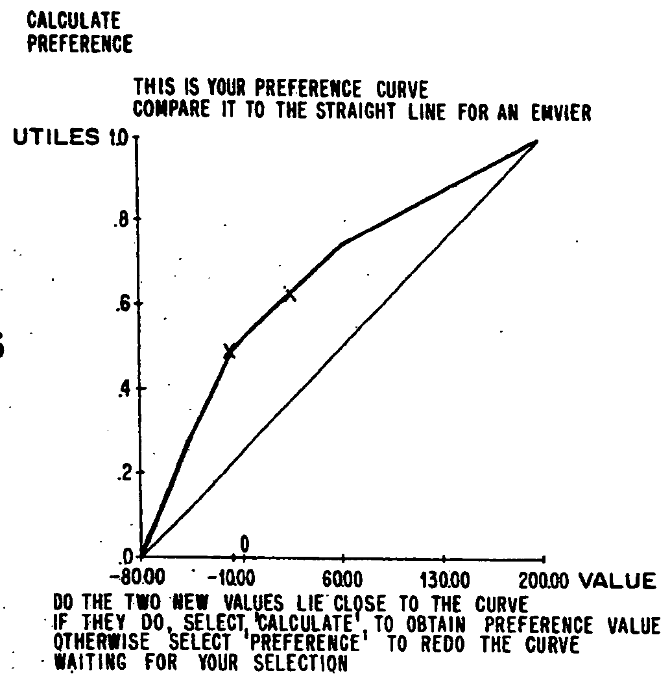

CALCULATE
PREFERENCE
THIS IS YOUR PREFERENCE CURVE
COMPARE IT TO THE STRAIGHT LINE FOR AN EMVIER
UTILES 1.0
.8
.6
.4
.2
.0
0
-80.00
-10.00
60.00
130.00
200.00 VALUE
DO THE TWO NEW VALUES LIE CLOSE TO THE CURVE
IF THEY DO, SELECT 'CALCULATE' TO OBTAIN PREFERENCE VALUE
OTHERWISE SELECT 'PREFERENCE' TO REDO THE CURVE
WAITING FOR YOUR SELECTION

FIG. 16

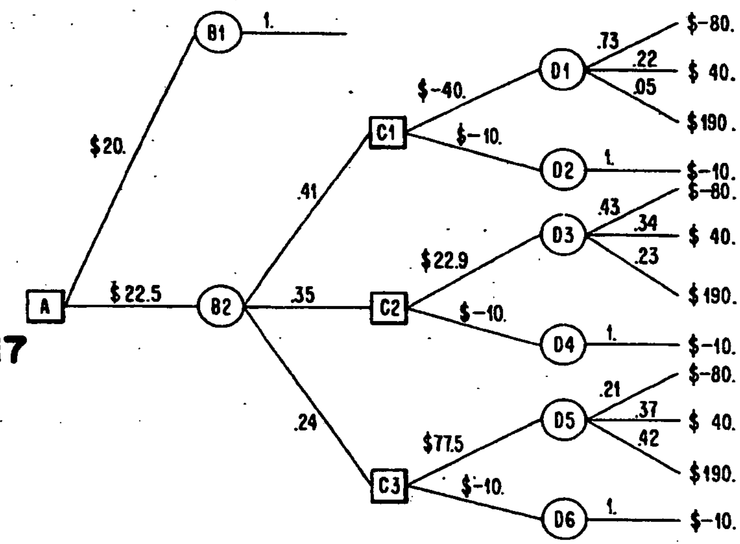
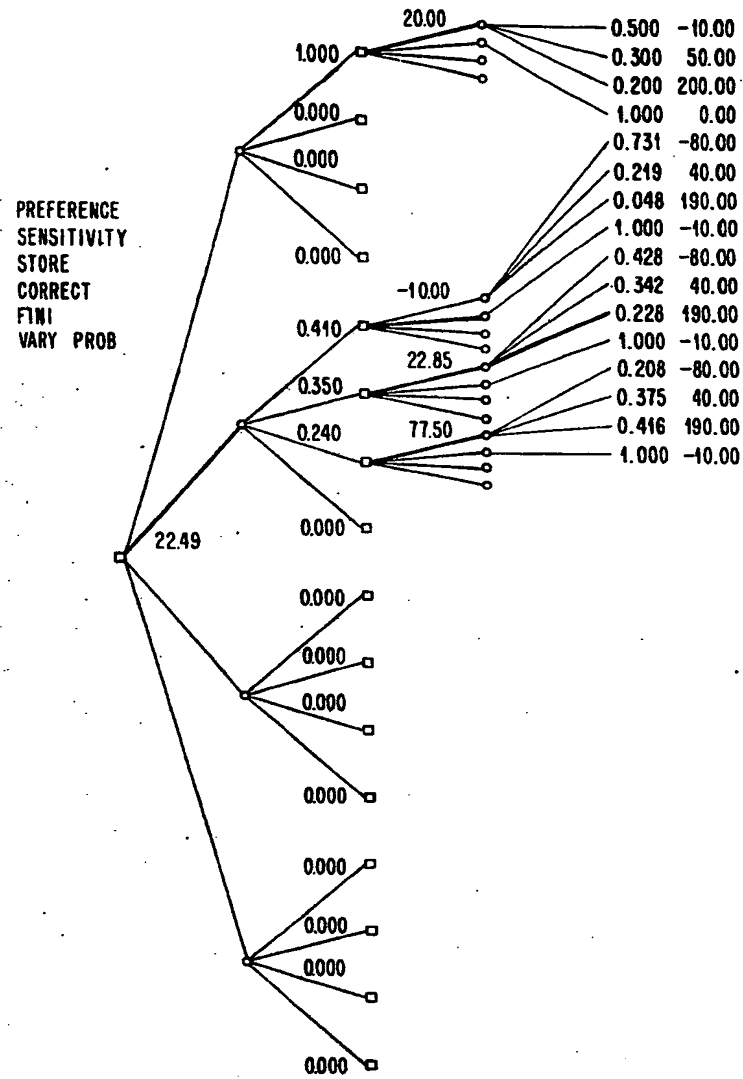
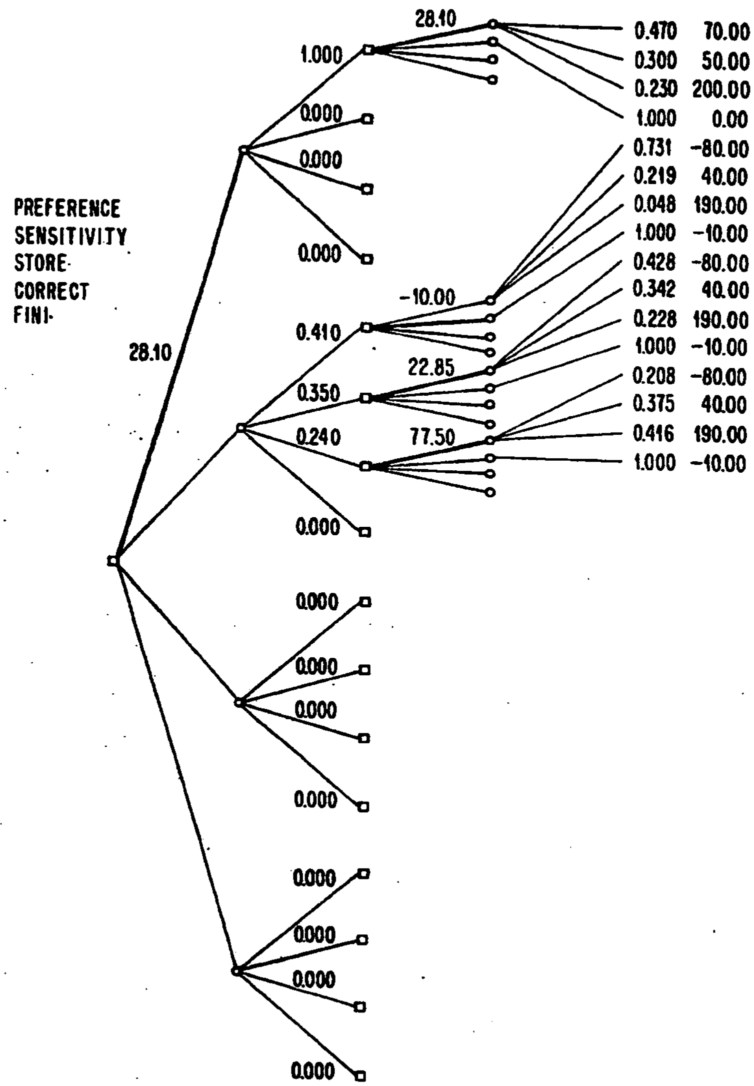

B1
1.
$20.
A
$22.5
B2
.41
.35
.24
C1
C2
C3
$-40.
$-10.
$22.9
$-10.
$77.5
$-10.
D1
D2
D3
D4
D5
D6
.73
.22
.05
1.
.43
.34
.23
1.
.21
.37
.42
1.
$-80.
$ 40.
$190.
$-10.
$-80.
$ 40.
$190.
$-10.
$-80.
$ 40.
$190.
$-10.

FIG. 17



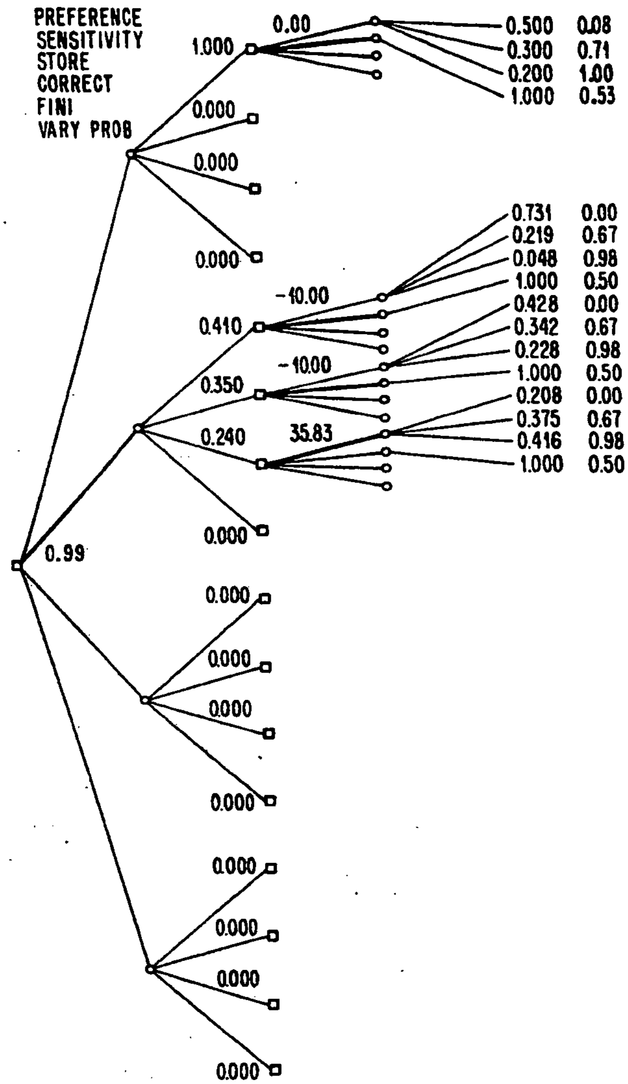

IF THE PROB OF BRANCH 83 DECREASES BY 0.026 FROM 0.228
YOUR FIRST DECISION CHANGES TO BRANCH A 1. YOUR VALUE IS 20.00
THE SAME PROB IS ALSO MOST SENSITIVE IN CHANGING VALUE

**FIG. 18**




I HAVE COMPUTED THE VALUE OF YOUR PROPOSITION
THE MOST VALUABLE CHOICES ARE PRINTED AND SELECTED BRANCHES ARE HEAVY
SELECT PREFERENCE TO DO A UTILITY ANALYSIS
SENSITIVITY FOR SENSITIVITY ANALYSIS, FINI WHEN FINISHED, PLEASE SELECT

**FIG. 19**




THE PREFERENCE VALUE OF YOUR PROJECT IS 0.999
WAITING FOR YOUR SELECTION FROM THE MENU

FIG. 20

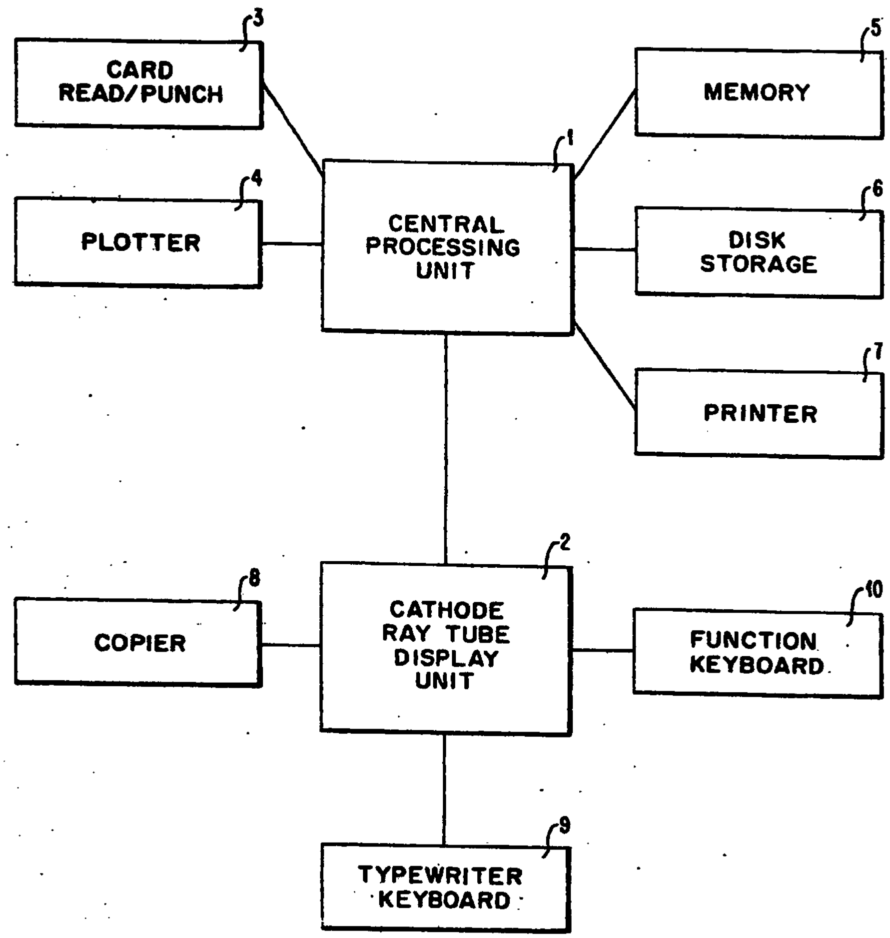

CARD READ/PUNCH
3
PLOTTER
4
CENTRAL PROCESSING UNIT
1
MEMORY
5
DISK STORAGE
6
PRINTER
7
COPIER
8
CATHODE RAY TUBE DISPLAY UNIT
2
FUNCTION KEYBOARD
10
TYPEWRITER KEYBOARD
9

FIG. 21